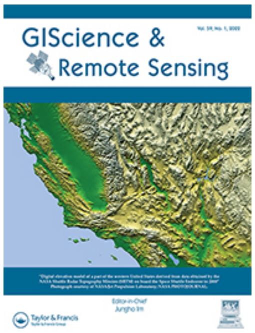

GIScience & Remote Sensing

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tgrs20

AutoML for estimating grass height from ETM+/OLI data from field measurements at a nature reserve

Nuno César de Sá, Mitra Baratchi, Vincent Buitenhuis, Perry Cornelissen & Peter M. van Bodegom

To cite this article: Nuno César de Sá, Mitra Baratchi, Vincent Buitenhuis, Perry Cornelissen & Peter M. van Bodegom (2022) AutoML for estimating grass height from ETM+/OLI data from field measurements at a nature reserve, GIScience & Remote Sensing, 59:1, 2164-2183, DOI: 10.1080/15481603.2022.2152304

To link to this article: https://doi.org/10.1080/15481603.2022.2152304

GISCIENCE & REMOTE SENSING
2022, VOL. 59, NO. 1, 2164–2183
https://doi.org/10.1080/15481603.2022.2152304

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS Check for updates

AutoML for estimating grass height from ETM+/OLI data from field measurements at a nature reserve

Nuno César de Sá[a], Mitra Baratchi[b], Vincent Buitenhuis[b], Perry Cornelissen[c] and Peter M. van Bodegom[a]

[a]Institute of Environmental Sciences (CML), Leiden University, Leiden, The Netherlands; [b]Leiden Institute of Advanced Computer Science (LIACS), Leiden University, Leiden, The Netherlands; [c]State Forestry Service, Amersfoort, the Netherlands

ABSTRACT
Remote sensing (RS) is now a standard tool used for grassland monitoring thanks to the availability of data at an unprecedented spatial and temporal resolution. The approaches to monitor grasslands often rely on the use of vegetation indices (e.g. NDVI) and empirical models trained on field data collected in tandem with the RS data. The best combination of models and features is often found by ad-hoc experimentation by the expert. This "classic" approach does not necessarily result in the best possible model. Automatic machine learning (AutoML) allows to automate this procedure by identifying the best possible pipeline in a data-driven manner. This study assesses the applicability of two distinct AutoML algorithms – AutoSklearn and AutoGluon – to monitor grass height from RS data and to systematically compare them to "classic" RS approaches. Grass height was estimated from Landsat ETM+ and OLI for a well-known conservation area as a case study. The "classic" RS approach emulated all possible ad hoc decisions by comparing all combinations of bands and vegetation indices against a naive use of the AutoML systems. While model selection and optimization are automated within AutoML models, for the "classic" RS approach, we used Bayesian optimization for hyperparameter tuning. We found that AutoML methods outperformed "classic" methods with the test error varying between ~1.73 cm ± 0.02 and ~1.78 cm ± 0.03 while for the "classic" methods it varied between ~1.84 cm ± 0.03 and ~2.81 cm ± 0.02. In the case of the "classic" methods, our exhaustive exploration of the possible feature combinations showed that while vegetation indices were always selected for the best models, which index got selected depended on the algorithm. The performance of AutoML compared to "classic" RS approaches clearly demonstrates the ability of these methods to quickly and effectively identify high-performing models. However, as this work focused on a single case-study, the results cannot be directly generalized to other study areas. Nevertheless, it provided a number of insights into future research opportunities to improve the use of AutoML in RS.

ARTICLE HISTORY
Received 31 May 2022
Accepted 9 November 2022

KEYWORDS
AutoML; machine learning; remote sensing; grassland; Landsat

Introduction

Remote sensing (RS) offers the ability to monitor ecosystem processes and services (Pettorelli et al. 2018; Shih, Stow, and Tsai 2019; Belgiu and Drăgut 2016) and is a natural choice for monitoring due to its ability to monitor at high spatial and temporal resolution (Reinermann, Asam, and Kuenzer 2020; Craglia et al. 2017). RS has been applied to various parameters and types of ecosystems such as to monitor above ground biomass (AGB) (Yang et al. 2018), functional traits (Wang et al. 2019b), species (Marcinkowska-Ochtyra et al. 2018), ecological invasions (Schulze-Brüninghoff, Wachendorf, and Astor 2020), or livestock forage (Wijesingha et al. 2020).

In general, RS approaches to monitor AGB rely on the use of well-established relationship between vegetation indices (e.g. normalized difference vegetation index or NDVI) and either opt to use generalized linear models (GLM) or machine learning regressions (MLR) to predict biomass (Xie et al. 2009). Some of the most common MLR algorithms used for this task are support vector machines (SVM) (Wang et al. 2019a), random forests (RF) (Mutanga, Adam, and Cho 2012; Wang et al. 2016; Wang et al. 2019a), K-nearest neighbors (KNN) (Dusseux et al. 2014; Zhu et al. 2017), Bayesian regressions (Tang, Ali, and Feng 2020; Xie et al. 2020), and artificial neural networks (ANN) (Taravat, Wagner, and Oppelt 2019). The high variety in methodologies, algorithms, and model optimization (if any) between the various applications to estimate biomass hampers its reproducibility.

CONTACT Nuno César de Sá n.q.cesar.sa@cml.leidenuniv.nl

This lack of reproducibility is a problem in the field of RS (Balz and Rocca 2020; Frery, Gomez, and Medeiros 2020; Colom et al. 2020). In particular, when using machine learning algorithms, there is a lack of precise description of either pre- and post-processing steps or model hyperparameters. Another factor complicating reproducibility is the common use of indices (Huang et al. 2021; Pôças et al. 2020; Guerini Filho, Kuplich, and Quadros 2020) as surrogate features alongside or instead of using the spectral bands to represent expected or well-known relationships between spectral properties and AGB (Gao et al. 2020; Derakhshan, Cutter, and Wang 2020). Many of the decisions on the selection of bands and/or vegetation indices are based on ad-hoc experimentation by the expert which does not always result in the best possible or most parsimonious models (Stromann et al. 2019; Maxwell, Warner, and Fang 2018; Georganos et al. 2018). Likewise, the use of machine learning procedures is highly dependent on machine learning knowledge which hinders their generalization, as well as expending valuable resources and time (He, Zhao, and Chu 2021; Maxwell, Warner, and Fang 2018; Khatami, Mountrakis, and Stehman 2016). Often there is no guarantee that the best possible machine learning pipeline is selected given that the RS expert might be biased toward the algorithms and processing pipelines that have been successful previously. Less optimal choices in terms of the choice of indices, machine learning algorithms or their hyperparameters may not only affect reproducibility but also the applicability of the same algorithm in a different location or task (Yang and Shami 2020; Colom et al. 2020). Considering the reproducibility crisis (Baker, 2016) and increasing efforts in the scientific community for making both data and software more easily findable and accessible (Wilkinson et al. 2016), these are aspects that need to be better accounted for.

One approach toward reproducibility is to automate the parts or the entire procedure by using automated machine learning (AutoML) (Xin, Zhao, and Chu 2021). These approaches have been shown to outperform humans/experts in competitions and to offer high quality results with minimal machine learning expertise (Xin, Zhao, and Chu 2021; Hanussek, Blohm, and Kintz 2020). Most advanced AutoML approaches focus on solving the so-called Combined Algorithm Selection and Hyperparameter Optimization (CASH) problem (Thornton et al. 2013) which, as the name motivates, automates the search of the best algorithm and its hyperparameters with the objective of minimizing the error of the task at hand.

There are a number of different AutoML systems available such as AutoSklearn (Feurer et al. 2015), H2O (LeDell and Poirier 2020), TPOT (Olson et al. 2016), Auto-WEKA (Chris et al. 2013) and AutoGluon (Erickson et al. 2020) which vary not just in the strategy to optimize the models but often are specific to the underlying machine learning library (e.g. AutoSklearn is built for scikit-learn). Currently, there are no “off-the-shelf” AutoML systems specific to RS applications, but there is an increasing interest on adapting these methods for RS (Salinas et al. 2021). While AutoML has recently been explored for RS applications (Salinas et al. 2021; Tedesco et al. 2021; Koh, Spangenberg, and Kant 2021; Li et al. 2022), there has not been a direct comparison against the “classic” RS approach emulating the decision-making of an expert.

Our research objective is to compare “classic” RS approaches, using vegetation indices, for estimating AGB against the AutoML approach. For this, we focused on a case study on the well-known Oostvaardersplassen nature reserve (OVP) (Staatsbosbeheer 2021) in the Netherlands by using field measurements of grass height and remote sensing data from the Landsat Enhanced Thematic mapper (ETM+) and the Operational Land Imager (OLI) data.

This research provides one of the first insights into the potential of AutoML systems for remote sensing and discusses the possibility of adapting these systems to specific remote sensing tasks by integrating field expertise with more advanced machine learning approaches.

Methods

Overview

The general approach of this research is visually summarized in Figure 1. The first step consisted of gathering grass height field data and acquiring all possible Landsat ETM+ and OLI observations that were within 14 days of the sampling events and to filter any data based on the QA_Pixel band to select only pixels captured under clear sky conditions.

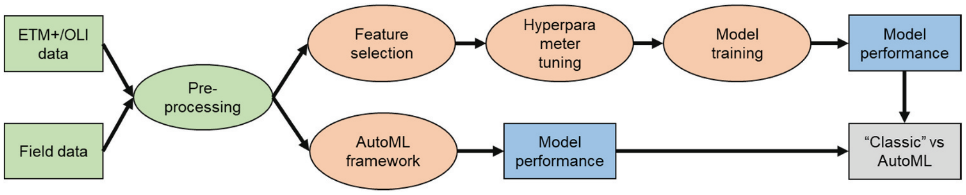

Figure 1. Overview of the research approach. The green color refers to minimal preprocessing steps such as data selection and extraction of the pixel values for each field sample. Orange shapes denote two main modeling approaches: "classic" (above) and AutoML (below). Blue refers to the model performance evaluation using a repeated k-fold approach. Finally, gray refers to the comparison between the results from both approaches.

The second step consisted of preparing the data to be used in the machine learning pipelines both for both approaches. We compared the AutoML approach against the expert-based "classic" approach by considering all possible combinations of bands and vegetation indices commonly used for the purpose of AGB monitoring. For this step, a sample of 50% of the data was used for hyperparameter tuning of the "classic" approach, as well as for developing pipelines in AutoSklearn.

For the model performance evaluation, we used an 80/20 approach with 20% of the data never being used on any of the model training procedures. The mean absolute error (MAE) distribution was calculated from the training and test data using a repeated *k*-fold cross-validation approach with 100 repetitions of 10 *k*-folds which has been shown to provide a better estimate of the error and variance than bootstrapping or just a single cross validation (Kim 2009). To ensure the maximum comparability between all these different steps of hyperparameter tuning, pipeline creation and repeated cross validation, the same data was used for all models. Finally, the distribution of the MAE of each model approach was obtained and the performance of the models compared.

Study area and data sources

Study area and field data

The Oostvaardersplassen nature area is managed by Staatsbosbeheer (Staatsbosbeheer 2021). The area is grazed by cattle, horses, and red deer which were introduced in the 1980s and 1990s (Cornelissen et al. 2014). The area is fenced off and there are no large predators. Up to 2018, the large herbivore numbers were controlled by food supply, severity of the winter and competition. Under these conditions, the large herbivores altered the vegetation in a dramatic way (Cornelissen et al. 2014). Within a period of 15 years, the diverse landscape, consisting of grasslands, tall herbs, reed, shrubs, and trees, was transformed into a homogeneous landscape dominated by short grazed grasslands. As a result, biodiversity decreased. In 2018, the management of the large herbivores changed. The numbers of the large herbivores are controlled to a lower level in order to create a diverse landscape and to offer room for more wetland-related bird species. To understand the effects of the changes in management not only on wetland birds but also on the large herbivores, there is a need for monitoring the available biomass (Cornelissen et al. 2014). Grassland biomass is monitored in the Oostvaardersplassen to ensure that enough food is available for the resident large herbivores and wintering geese and is collected by measuring sward height in preset transects (shown in Figure 2) which were used for this research. Samples were collected in intervals of 30 or 50 m depending on the transect. Our data were collected between May 2013 and November 2017 but not all could be used due to either sky conditions or lack of enough high-quality remote sensing data. Table 1 shows a summary of the dates where the RS acquisitions aligned with the fieldwork data. For measuring the grass height, a representative sample was selected at each sampling location and grass height was measured using a disc except when the presence of tall herbs (e.g. thistles and ragworts) impeded a proper measurement in which case we used a ruler to directly measure the grazed grass height.

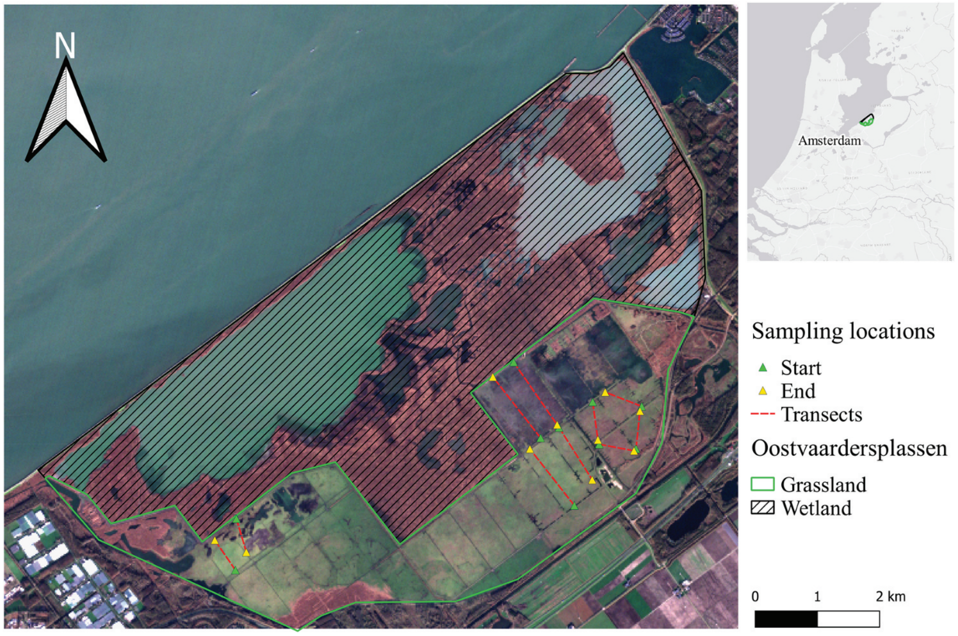

Figure 2. Study area showing the division between the grassland and wetlands of the Oostvaardersplassen. Field data were collected along pre-defined transects with vegetation height being measured every 30 or 50 m.

Table 1. Summary of the data used in this research.

Field sample date	Field/RS samples
07/05/2013	137
07/06/2013	170
08/07/2013	112
09/08/2013	206
12/09/2013	5
16/01/2015	125
16/04/2015	139
03/06/2015	164
10/07/2015	356
21/12/2015	141
16/09/2016	215
08/11/2016	37
14/06/2017	327
25/07/2017	117
28/08/2017	14
01/11/2017	162
Total samples:	2427

Not all field samples could be used due to weather or and/or missing remote sensing data due to the scanline error of Landsat ETM+. This table shows how many samples were available for both field and RS for each field sampling date.

Remote sensing data

We opted to use Landsat ETM+ and OLI, because their long-term availability and temporal resolution were sufficient for our purpose. Transect locations were transferred to a shapefile and then used to extract the pixel values from Landsat. All Landsat ETM+ (on board Landsat 7) and OLI (on board Landsat 8) surface reflectance data products were collected from the Google Earth Engine (Gorelick et al. 2017). Both Landsat 7 and 8 have a 16-days repeat cycle at the equator, but when used together, the repeat cycle is halved at 8 days (Murphy et al. 2016).

Given that there are slight differences in terms of the wavelengths as seen in Appendix 1, we opted to use only those bands which are equivalent between both sensors. Satellite observations closest in time to the field sampling data were collected using a time window of ± 15 days. These were filtered to consider only one satellite observation per field sample per time taking also in consideration the quality flags for cloud cover and scanline error in the case of Landsat ETM+. Coincidently, the study area is at the intersection between two different Landsat overpass rows: 024 and 023 and, arbitrarily, we opted for row 024 and only used row 023 when row 024 was not available, yielding a total of 2427 samples of which 1943 (~80%) were used for training and 484 (~20%) were used for validation (see Table 2).

"Classic" approaches commonly use individual bands (as selected above) and vegetation indices, based on combinations of bands. We tested four indices that are commonly used for AGB estimation

Table 2. Summary of total RS samples used per each sensor for both training and validation of the models.

	ETM+	OLI	ALL
Training	1408	535	1943
Validation	351	133	484
Total	1759	668	2427

(Garroutte, Hansen, and Lawrence 2016; Huang, Chen, and Cosh 2009; Ullah et al. 2012) alongside B1 to B5 and B7 (B2 to B8 in OLI):

$$\text{Normalized Difference Vegetation } Index\text{ (NDVI)} = \frac{B4 - B4}{B4 + B4}$$

$$\text{Enhanced Vegetation Index (EVI)} = \frac{B4 - B4}{B4 + 6 \times B3 - 7.5 \times B2 + 1}$$

$$\text{Soil Adjusted Vegetation Index (SAVI)} = \frac{B4 - B3}{B4 + B3 + 0.5} \times (1 + 0.5)$$

$$\text{Normalized differencewater index (NDWI)} = \frac{B4 - B5}{B4 + B5}$$

where B5, B4, B3, and B2 correspond to the shortwave infrared, near-infrared, red, and green bands of the ETM+/OLI sensors. For more details on each band, see Appendix 1, and the final dataset used in this research is made available in Appendix A2.

Algorithms

"Classic" approach

This approach for pixel-based classification refers to the common methods that are used in remote sensing consisting of testing single and/or multiple models with hyperparameter tuning and expert-based feature selection (Reza, Mountrakis, and Stehman 2016; Maxwell, Warner, and Fang 2018). To emulate how expert RS scientists approach the problem of AGB estimation (Chen et al. 2019; Zhang et al. 2020b; Ali et al. 2015; Phiri et al. 2020), we opted to test some of the more commonly used algorithms used with a focus on optical data (Ji et al. 2021; Pham et al. 2020; Tang, Ali, and Feng 2020; Taravat, Wagner, and Oppelt 2019; Phiri et al. 2020).

The algorithms selected were generalized linear models (GLM) (Barrachina, Cristóbal, and Tulla 2015; Marildo, Mora Kuplich, and De Quadros 2020), support vector machines (SVM) (Wang 2019; L. Chen et al. 2019), random forests (RF) (Chen et al. 2018; Wang et al. 2016), k-nearest neighbors (KNN) (Dusseux et al. 2014; Zhu et al. 2017), Bayesian regression (BR) (Xie et al. 2020; Tang, Ali, and Feng 2020), and artificial neural networks (ANN) (Vafaei et al. 2018; Yang et al. 2018).

AutoML approach

Two AutoML pipelines – AutoSklearn (Feurer et al. 2015) and AutoGluon (Erickson et al. 2020) – were chosen because they represent different approaches to deal with machine learning automation: AutoSklearn aims to select and identify the best pipeline(s) for a specific machine learning application (Feurer et al. 2015) while AutoGluon uses sequential stacking of machine learning algorithms which can minimize the need for hyperparameter tuning (Erickson et al. 2020). In the next section, we provide a more detailed explanation of both these frameworks.

AutoSklearn

AutoSklearn (Feurer et al. 2015) is an AutoML pipeline built on top of the scikit-learn API (Pedregosa et al. 2011) that uses Bayesian optimization for hyperparameter tuning and algorithm selection. The system consists of three main components: (1) meta-learning, (2) machine learning pipeline, and (3) ensemble construction (see Figure 3). The meta-learning component of AutoSklearn consists of starting the model pipeline with parameters that were tested in other similar datasets (Feurer et al. 2015). Meta-learning, therefore, assists in finding the optimal solution quicker by reducing the hyperparameter space to be tested, being more robust against random effects and with a lower variance of the error.

The machine learning pipelines automatically configured by AutoSklearn consist of a combination of three different tasks: data preprocessing; feature preprocessing; algorithm selection & hyperparameter tuning (Feurer et al. 2015). In terms of data preprocessing, AutoSklearn uses methods built in scikit-learn for preprocessing data (e.g. one-hot encoding, imputation, rescaling) (Pedregosa et al. 2011).

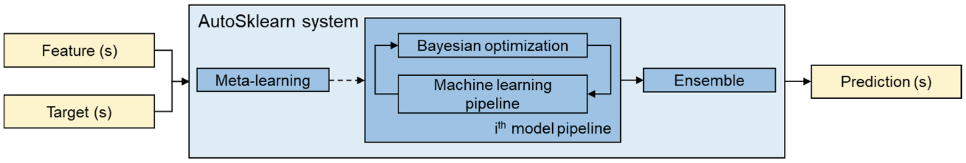

Figure 3. Overview of the autosklearn automl system adapted from Feurer et al. (2015). The use of meta-learning to speed up the fitting procedure is optional and the proposed ensemble weights are calculated based on an ensemble approach proposed by Caruana et al. (2004).

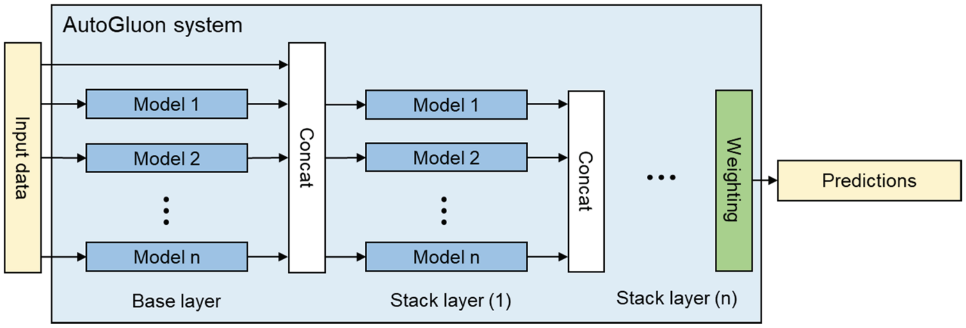

Figure 4. AutoGluon multilayer stack ensemble strategy adapted from Erikson et al. (2020). The first base layer uses the input data and the models as input to the next stack layer. This process is repeated in *n* successive layers. Hyperparameter tuning is not recommended to avoid overfitting.

Regarding the feature preprocessing, AutoSklearn tests several procedures that potentially improve the fitness of the models such as decomposition methods (e.g. PCA, SVD), polynomial expansions, feature clustering, among others (see Suppl. Materials Feurer et al. 2015 for an extensive description). The algorithm selection tests the 15 base scikit-learn algorithms (Pedregosa et al. 2011) including searching for the best possible hyperparameter for each. Finally, AutoSklearn uses a model ensemble approach (Feurer et al. 2015; Caruana et al. 2004) which consists of leaving part of the data out as validation set during training to internally test and validate new model pipelines iteratively. Their ensemble weight is based on their impact on the chosen evaluation metric. All analyses are based on AutoSklearn version 0.11.1.

While it is apparent that AutoSklearn has the potential to be more computationally intensive than the "classic" approach, this might not be necessarily true. The entire model pipeline is optimized using a Bayesian optimization approach and – although many aspects of its processing can be customized – the total time spent in optimization is the only parameter that the user needs to set (time_left_for_this_task). We configured the total time spent in training with increasing time (30, 180, 300, and 600 s) which results in more time being available for testing different model pipelines. The implication of using Bayesian optimization is that not all models/configurations have to be tested but instead AutoSklearn is able to quickly focus on the models/configurations with more potential improvement instead of having to test all (or a very large number of) model/configurations. We ran AutoSklearn with and without meta-learning to test how using the model library already available with the algorithm did improve (or not) the model performance.

AutoGluon

AutoGluon was developed as an open-source project by the Amazon Web Services and it approaches the AutoML problem from a different perspective compared to most of its predecessors (Erickson et al. 2020). The main difference is in the use of sequential stacking of various machine learning models with limited hyperparameter tuning (Figure 4). The first layer (base) is composed of the models trained on the input data using a predefined sequence of models which are then used as features for the next layer (Figure 4). This process is repeated *n* times depending on the time budget and preset configuration by the user. This AutoML framework is fundamentally based on a multilayer stack ensemble approach to perform better than single models (van der Laan, Polley, and Hubbard 2007). The algorithms currently used within AutoGluon are neural networks, lightGBM, catboost, random forests, xgboost, and k-nearest neighbors which is a significantly smaller number of algorithms than what is available in AutoSklearn. Finally, the ensemble selection approach consists of selecting models based on their impact on performance (Caruana et al. 2004) as AutoSklean also uses. To avoid overfitting of the models, AutoGluon uses a repeated k-fold bagging approach on a validation dataset and the model hyperparameters can be automatically fine-tuned using Bayesian optimization. In this case, the user can also configure many aspects of the model such as selecting only specific models or adding custom instances but the main parameter relates to the total time being used for training the models (time_limit) which is the total time provided for AutoGluon to find an ensemble. Because AutoGluon trains model pipelines in succession (with hyperparameter even being optional), it can find high performance results even with small computational effort. On the other hand, also because of the approach of successively training models in a preset order, it is more of a black-box approach.

Experimental design

The dataset was randomly divided into a training (80%) and test dataset (20%) as shown in Figure 5. The cross-validation used 10 folds only of the training data and was repeated 100 times with different random samples of the training data to ensure a good estimate of the mean error and is adapted from commonly used approaches (Wang et al. 2019a). This means that in each iteration, nine folds were used for training and the remaining fold was used to estimate the training error while the test error was performed on the left-out dataset (20%) (see Figure 5). This random sampling of the data was performed only once to ensure that every model used the same data for both training and validation and thus ensure a fairer comparison between the models.

We tested all possible feature combinations for both the GLM and MLR approaches – as may be selected by an expert – against the AutoML algorithms where no decision regarding feature selection is made. The total number of possible model structures was defined by all combinations of six bands and four vegetation indices, leading to a total 1023 possible model structures.

To limit the negative effect of autocorrelation on GLM models, multicollinearity was evaluated through a combination of the Pearson's correlation and Variance of Inflation Factor (VIF) and only model structures with a lenient VIF <10 were tested (Stine 1995). As a consequence, only 165 of the total possible 1023 models were deemed acceptable in the case of GLM. For the remaining models in the "classic" approach each of the 1023 possible feature combinations was tested.

In the case of MLR, hyperparameter tuning was performed using Bayesian optimization for 50 iterations for each of the 1023 possible feature combinations. We found no specific recommendation on the number of iterations that should be used, but during the initial tests, we observed that model performance stopped improving around between 20 and 30 iterations for most models and therefore we considered 50 iterations to be sufficient. This procedure was implemented using the scikit-optimize (0.8.1) package. The hyperparameter space is given in Appendix A3 and was based on previous RS research or, if no clear guidelines were available, based on non-RS research (Yang and Shami 2020; Belgiu and Drăgut 2016; Taravat, Wagner, and Oppelt 2019; Zhuo et al. 2016). A single random subsample of 50% of the data was used for hyperparameter tuning of each independent model structure. Model performance was based on the mean absolute error (MAE) based on the test dataset with the Mann–Whitney *U* test being used to test if there was a significant difference between the accuracy of the best models.

Results

Classic vs AutoML results

The MAE of only the best models is shown in Table 3 while the final results for all possible model features

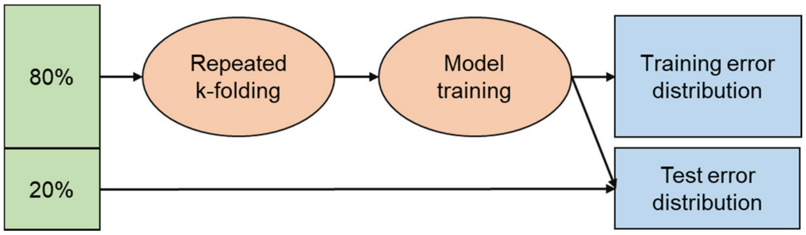

Figure 5. Overview of the experimental setup. 80% of the training data was used to generate a repeated *k*-fold estimate of the training error (100 repetitions of 10 folds, one being left out for training error estimates). The remaining 20% of the data was not used for any training procedure. To ensure comparison, the same data were used for each different model and iteration.

Table 3. Best overall models for each approach.

Approach	Algorithm	Model Structure / Time budget	Error type	MAE (cm)	σ
Classic	Generalized linear model	NDWI+B1+B2	Test	2.71	0.02
			Training	2.99	0.26
	Support vector machine	NDVI+EVI+SAVI+NDWI+B2	Test	2.38	0.04
			Training	2.65	0.28
	Random forest	B1+B2+B3+B4+B5+B7	Test	1.99	0.01
			Training	2.22	0.22
	K-nearest neighbour	EVI+B1+B2+B3+B5	Test	1.84	0.03
			Training	2.12	0.23
	Bayesian regression	NDVI+EVI+NDWI+B1+B2+B3+B4+B5+B7	Test	2.62	0.01
			Training	2.94	0.24
	Artificial neural network	EVI+SAVI+B1+B2+B3+B4+B5	Test	2.81	0.02
			Training	3.09	0.26
AutoML	AutoGluon	600s	Test	1,78	0,03
			Training	1,83	0,20
	AutoSklearn (with meta-learning)	300s	Test	1,75	0,01
			Training	1,83	0,21
	AutoSklearn (without meta-learning)	600s	Test	1,73	0,02
			Training	1,84	0,21

The green shading highlights the best overall model which is AutoSklearn without meta-learning and with 600 s of training time.

and time budgets is provided in Appendix D. Both AutoML approaches had better accuracy than any of the "classic" approaches with AutoSklearn (without meta-learning) being the best overall with ~1.73 cm mean absolute error and both AutoSklearn (with meta-learning) and AutoGluon closely behind with an error of 1.78 cm (Table 3). It should be noted that the mean grass height of the entire dataset is of ~5.08 cm and therefore the error margins obtained are still significant in relation to the actual grass height variation.

It was found that there was a significant difference between all the tested models ($p < 0.001$) which implies that the AutoML approaches were significantly better than the classic approach.

Classic approach

To emulate any decision made by a remote sensing expert, we tested all possible (and reasonable) combinations of features (e.g. spectral bands plus NDVI is commonly used). While this section provides a summarized version of the results, the entire dataset summarizing the model performance for both the classic and AutoML approach is provided in Appendix D.

The best model structures (and coinciding MAE) are shown for all linear models and classic machine learning algorithms in Table 3. Overall, the machine learning algorithms performed better than the linear models and showed consistently better predictive ability irrespective of the model structure.

The best overall GLM had the feature combination of NDWI+B1+ B2 and a test error of ~2.71 cm, while the 165th model which had the structure B1 + B5 had a test error of ~3.31 cm. Therefore, in the case of linear models, it appears that feature selection played a significant role in improving the model performance.

The best MLR was the KNN with the model structure EVI+B1+ B2+ B3+ B5 and a test error of 1.84 cm, while ANN yielded the worst results (Table 3). When compared to the linear model, only the RF and KNN performed better on the test set while GLM used a more parsimonious model with only two bands and one vegetation index.

Figure 6 provides an overview of the tested models and the variation in terms of the test error for the different feature combinations. Variation in MAE decreases for all models when selecting the best performing feature combinations. This suggests that although feature selection is important, multiple combinations of features can produce a similar result. Indeed, when looking at the best 25 models for each algorithm there were a total of 104 different feature combinations which were only three times shared by three different algorithms and one time shared by two different algorithms (Table 4). In all other cases, each feature combination was used by only one of the six algorithms.

Likewise, in the case of the best 50 models, out of 220 different feature combinations, only 58 times was

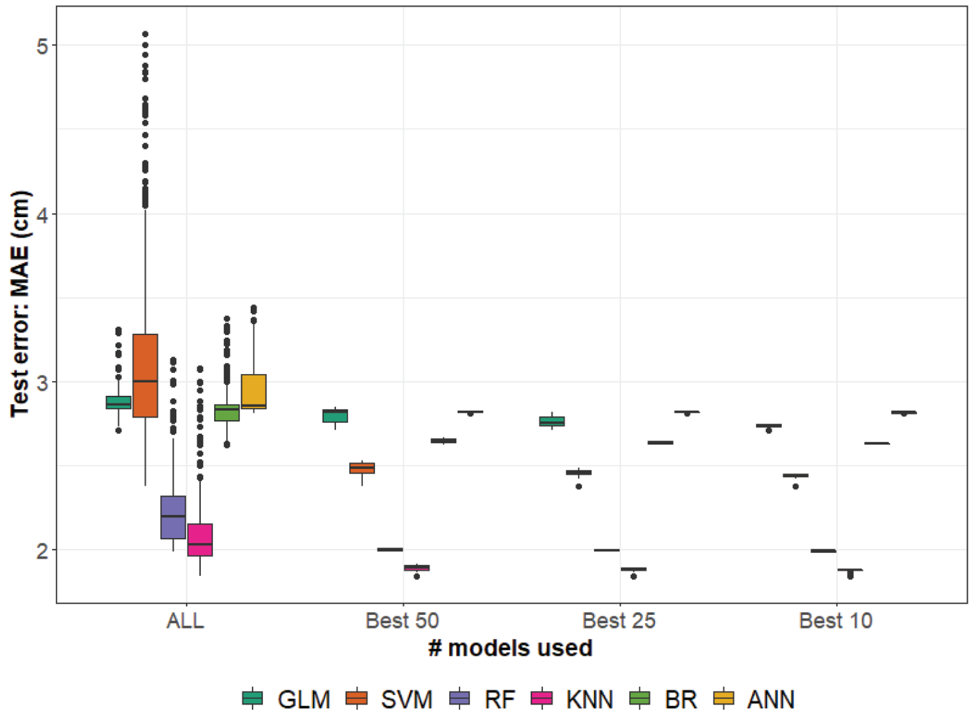

Figure 6. Distribution of the test error using different feature combinations: All, the best 50, 25 and 10. This visualization aims to summarize how much variation occurs depending on the model structures used. The data used for this plot is provided alongside this manuscript.

Table 4. Model structures shared by two or more algorithms for the 25 best results.

Model structure	ANN	BRR	KNN	RF	SVM
EVI+B1+ B2+ B3+ B4+ B5	x			x	
EVI+B1+ B2+ B3+ B5+ B7			x		x
EVI+NDWI+B1+ B2+ B3+ B4+ B5+ B7				x	x
NDVI+B1+ B2+ B3+ B4+ B5	x	x			
NDVI+EVI+B1+ B2+ B3+ B5	x	x			
NDVI+EVI+B1+ B2+ B3+ B5+ B7		x	x		
NDVI+EVI+NDWI+B1+ B2+ B3+ B5	x	x			
NDVI+EVI+SAVI+B1+ B2+ B3+ B4+ B5	x	x			
NDVI+EVI+SAVI+B1+ B2+ B3+ B5		x	x		
NDVI+EVI+SAVI+B1+ B2+ B5	x		x		
NDVI+EVI+SAVI+NDWI+B1+ B2+ B3+ B5	x	x			
NDVI+SAVI+NDWI+B1+ B2+ B3+ B4+ B5	x	x			
NDWI+B1+ B2+ B3+ B4+ B5				x	x
NDWI+B1+ B2+ B3+ B4+ B5+ B7				x	x
SAVI+B1+ B2+ B3+ B5	x		x		
EVI+B1+ B2+ B3+ B4+ B5+ B7	x			x	x
NDVI+EVI+SAVI+B1+ B2+ B3+ B5+ B7	x	x	x		
NDVI+EVI+SAVI+NDWI+B1+ B2+ B3+ B4+ B5+ B7	x	x	x		

the same combination used by two to four different algorithms (see Appendix D). In the case of the best 10 models, only once the feature combinations were shared between two different algorithms (see Appendix D). This implies that in the case of MLR, different models select different features and can lead to similar results. Therefore, interpreting that a specific feature (e.g. NDVI) plays a critical role in improving model accuracy can be the result of chance.

Figure 7 summarizes how commonly different features were selected by the different models for the 10, 25, and 50 best model structures. In general, features in the visible range were most often selected. In particular, the B2 (Green) was used almost 100% of the cases for GLM, RF, KNN, BR, and ANN. The exception to this was SVM, which did not show any preference for any feature. The use of vegetation indices was not associated with the best models in most cases, except for KNN where EVI was used over 85%

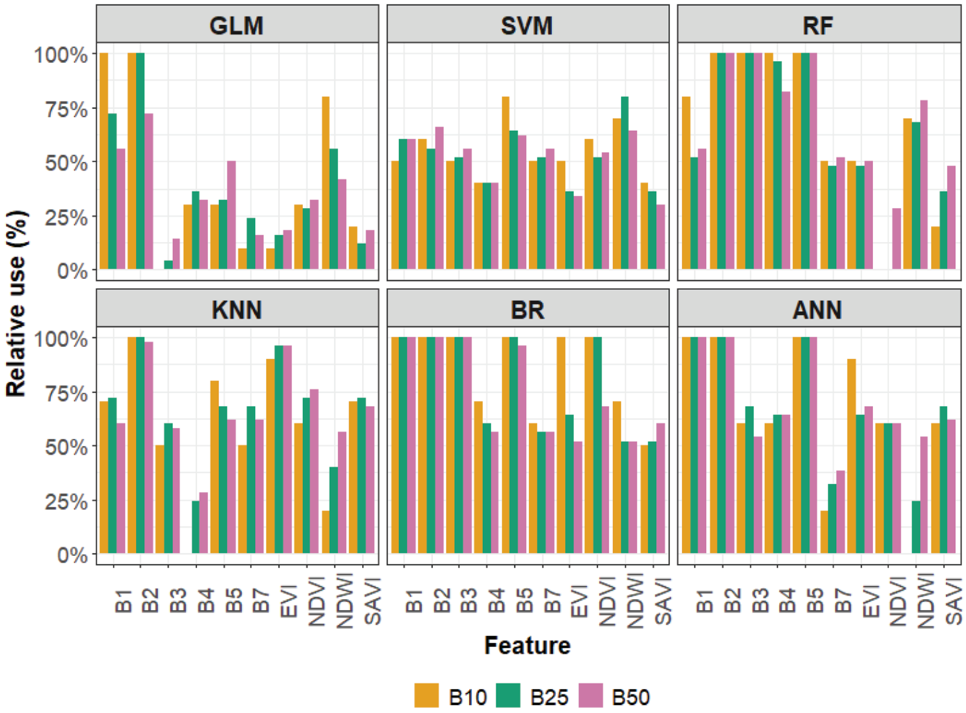

Figure 7. Relative use of each feature by the 10, 25, and 50 best model structures. The relative use represents the ratio between the number times a feature was used by the total possible number of model structures (10, 25, or 50).

of the time for the top 50 models and BR which used EVI or NDVI in its top 10 and top 25 best models.

AutoML frameworks

While both AutoML models produced the most accurate results, AutoSklearn needed more time to produce a model when compared to AutoGluon (Figure 8). Only AutoGluon was able to produce a model ensemble with 30 s of time budget for training. This is somewhat expected as while AutoSklearn relied on Bayesian optimization to address the CASH problem, AutoGluon attempts to avoid it by using a multilayer stack ensemble approach. In both cases, there is only minimal improvement/difference between the increasing time budgets. This implies that both frameworks can efficiently find a good set of models for estimating grass height from remote sensing data.

To further understand if there were significant patterns within AutoSklearn pipelines, we explored if there are any particular models and/or processing steps that are consistently selected (see Figures 9, 10, 11). This analysis is only relevant for AutoSklearn where Bayesian optimization is being used to define the best pipelines which is not the case in AutoGluon. In the latter, the approach is to use a multilayer stacked ensemble where models are trained in a predefined succession and retrained with minimal hyperparameter tuning to avoid overfitting.

For both AutoSklearn experiments, either when using meta-learning or not, tree-based algorithms such as extremely randomized trees, gradient boosting, decision trees, and random forests played a significant role. As the time budget increased, more algorithms were tested and their models were added to the ensemble. However, it is clear that when using meta-learning, there is a smaller variation in the ensemble weights with increasing time budgets (Figure 9). This is expected because meta-learning is based on a set of previously validated pipelines/parameters which are used to initiate the fitting procedure and guide the next steps of the Bayesian optimization. The best overall model pipeline was the one with a 600s time budget without meta-learning which was a combination of KNN with mostly tree-based algorithms (Figure 9, right). Given that the KNN also was the best performing algorithm from the classic approach, it seems that this model is particularly fit for this task.

Further exploration of the feature preprocessing methods applied by AutoSklearn in function of the

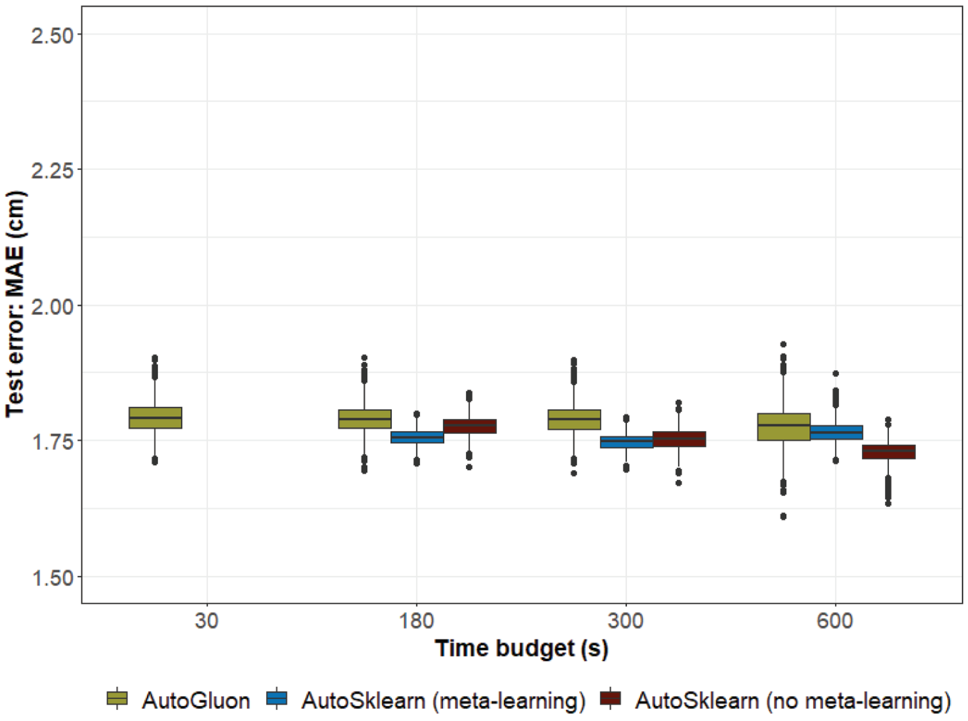

Figure 8. AutoSklearn and AutoGluon test error for increasing time budgets. For both AutoML frameworks we can observe that there is minimal increase in performance irrespective of allocated time budget for training. The exception is that only in the case of AutoGluon, a solution was obtained for the minimum time of 30 s.

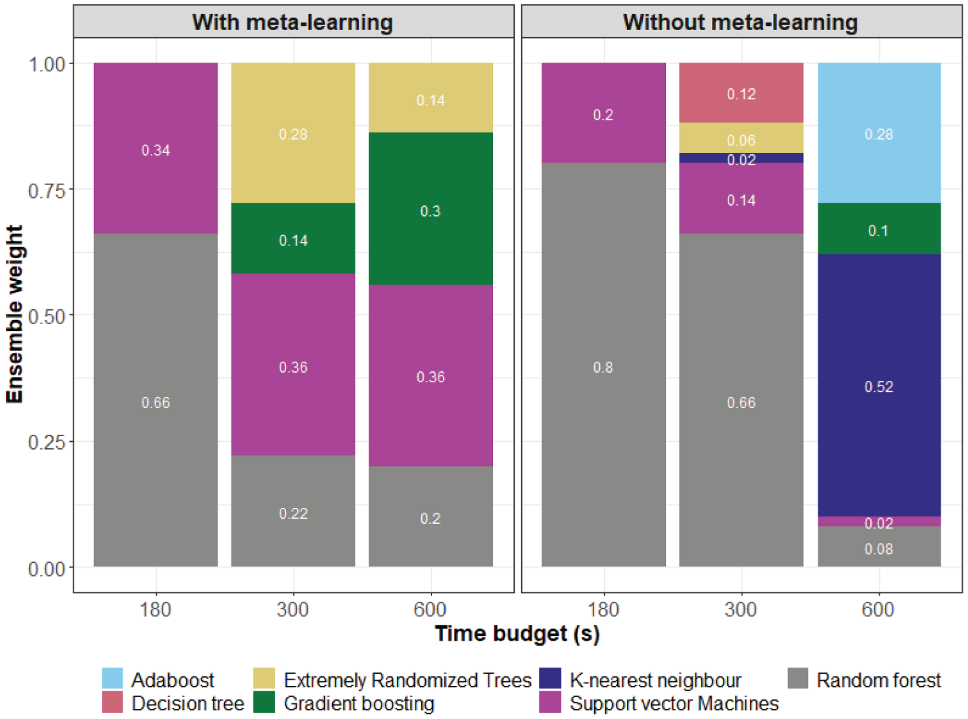

Figure 9. Aggregated ensemble wait attributed by AutoSklearn to each algorithm for each time budget. AutoSklearn can use multiple versions of the same algorithm with different sets of hyperparameters and preprocessing steps. The figure above shows the sum of weights attributed to each algorithm for visual simplicity.

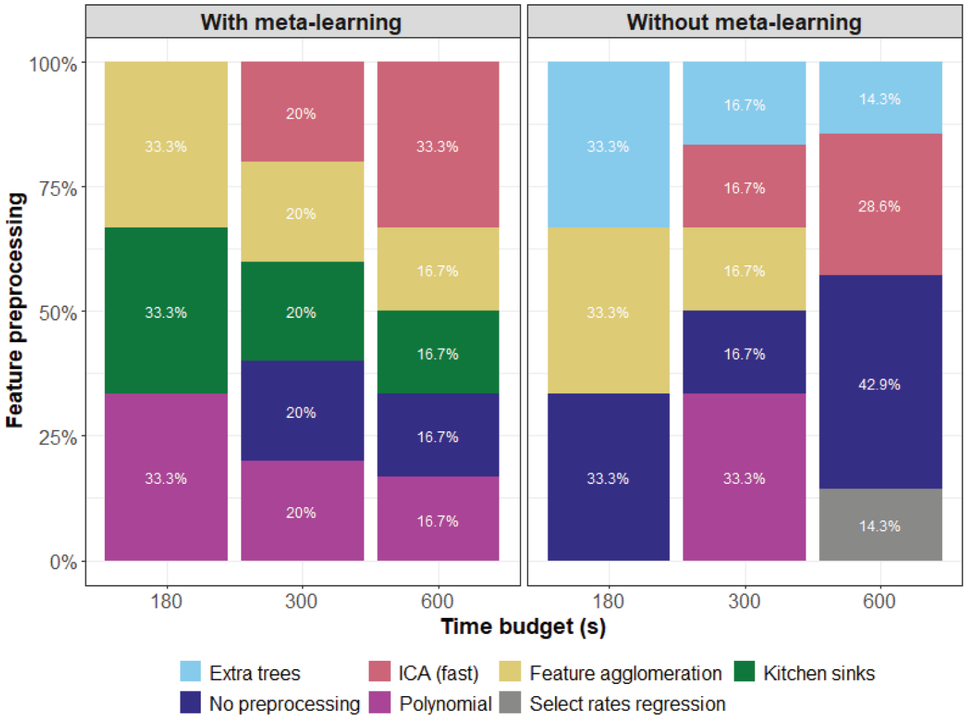

Figure 10. Aggregated feature preprocessing usage per time budget with and without meta-learning. Different settings of the same feature preprocessing can be used by AutoSklearn in different pipelines and these were aggregated by method and percentage of times used per time budget.

time budgets is shown in Figure 10, allowing a visual exploration of any patterns in preprocessing. Overall, no apparent preference for a specific feature preprocessing step was found (Figure 10) although without meta-learning resulted in more variation of between the algorithms being used given different time budgets. This is not unexpected given that the use of meta-learning implies that initial pipelines used to start the optimization procedure are similar for each time budget iteration.

To further investigate if there are specific patterns of data preprocessing steps in AutoSklearn we summarized the different algorithms being used in function of the time budget in Figure 11. Data preprocessing is divided into two groups of algorithms which are based on the feature type: categorical and numerical. In our case, the categorical data refer to the Landsat sensor used (ETM+ or OLI) and given that we used almost 3 times more ETM+ data than OLI data (see Table 2) it is not surprising that AutoSklearn tested the use of minority coalescence (Figure 11, top).

Data preprocessing applied to numerical data consists of imputation (to deal with missing data) and rescaling which applies scaling to the input values before proceeding. Imputation was excluded from our analysis given that our data was prepared so that there would be no missing values. Regarding rescaling, Figure 11 (bottom) shows once again that when not using meta-learning there is higher variability of the algorithms being tested. Nevertheless, in either case, there is no apparent reliance in any particular data preprocessing algorithm for rescaling even if there is higher diversity of algorithms being tested when not using meta-learning (Figure 11, bottom).

Discussion

Overall, our results showed that both AutoML algorithms performed significantly better than classic RS approaches in estimating grass height (Table 3). Although feature selection and model hyperparameter selection is commonly based on expert knowledge or previous experience in RS (Maxwell, Warner, and Fang 2018), we opted for an exhaustive test of all feature combinations and hyperparameter tuning to exemplify the "classic" RS approach. This allowed us to compare

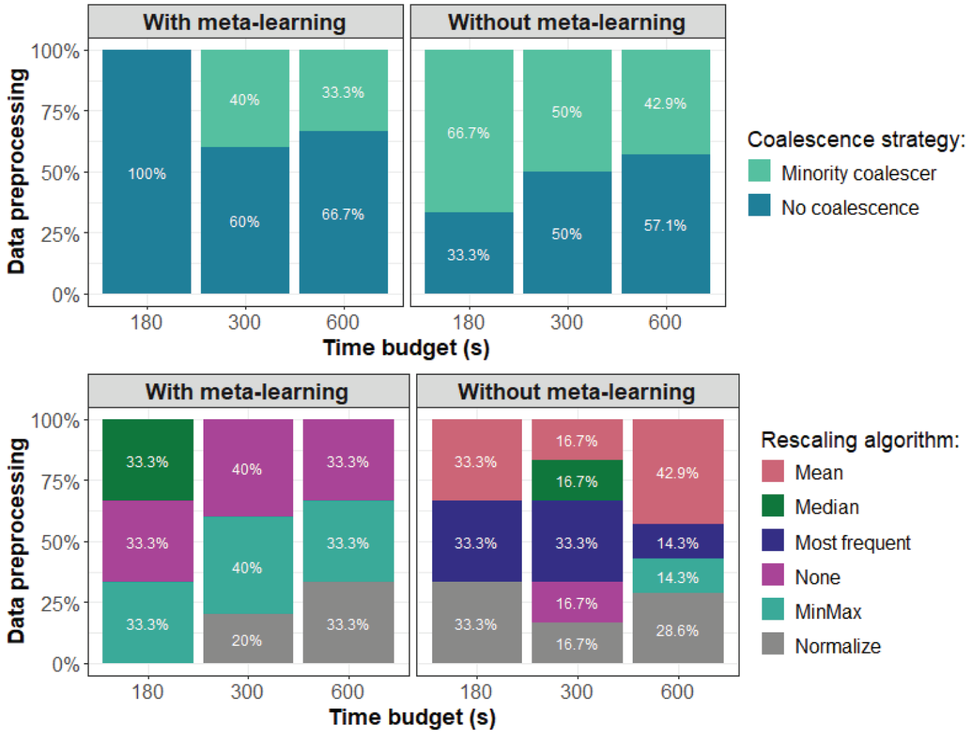

Figure 11. Aggregated data preprocessing for categorical (top) and numerical (bottom) inputs. As in the previous cases, different pipelines can use different versions of data preprocessing methods with different settings. These were aggregated into % of times used per time budget for visual simplicity.

the AutoML algorithms against all decisions made by an RS expert regarding feature selection and ensured that we selected the best possible model in terms of the mean absolute error. While AutoML approaches are hardly applied in RS (see recent examples for UAV phenotyping of wheat (Koh, Spangenberg, and Kant 2021) and neural architecture search (Zhang et al. 2020a; Peng et al. 2021; Wang et al. 2021), they have been shown to surpass human expertise in other fields (Hanussek, Blohm, and Kintz 2020). By simulating what could be expert decisions in our experiment, we also showed that AutoML surpassed human expertise for our case. Hence, there is an opportunity for further research and particularly on the adaptation of AutoML pipelines for RS tasks (Salinas et al. 2021).

Our exhaustive testing of the impact of feature selection for the classic approach showed that feature selection plays an important role in model performance (see Table 4 and Figure 7). In itself, this is not unexpected as feature selection has been shown to be significant for model performance previously (Georganos et al. 2018; Stromann et al. 2019), but in our case, we also observed a significant variation in the features being selected. This lack of consistency in the feature combinations can be partly explained by the algorithms themselves using different parts of the data. Nevertheless, more consistency between different models would be expected as the relationship between infrared bands and vegetation indices with AGB is well established in the field of RS (Zhu et al. 2017; Xie et al. 2009; Wang et al. 2016) even if there is a significant variation in terms of the most significant bands (Wen et al. 2020).

Furthermore, our results showed minimal differences in terms of performance between the best 50 feature combinations implying that any of these best 50 structures could be arbitrarily chosen without significant loss in performance. If the intention was to explore which bands/vegetation indices are contributing to a specific model, this would be problematic as a similar (or better) performing model with a different set of features can potentially be found. When summarizing the relative use of each specific feature, we found that some features were selected more often than others, with bands 1 to 5 being chosen most of the times (Figure 7). At the same time, while vegetation indices were also commonly used in the model structure, we found that they were selected much less often (see Figure 7). This can be seen as contradictory to previous research which selected vegetation indices based on the intuition they are the most useful to estimate AGB (Chen et al. 2018; 2019). On the other hand, it is expected that vegetation indices are highly correlated between themselves which can imply that one performing better than another for a given empirical experiment can be an arbitrary result.

The comparison of the two AutoML systems shows that while both AutoML frameworks produced the best results compared to the classic models, AutoGluon was able to produce a model quicker (see Figure 8). This is expected given that the AutoGluon strategy does not rely on testing different preprocessing steps or model hyperparameters but instead applies a stacking of weak learners approach which has been shown to produce highly accurate results (Dietterich 2000). Another difference between these two frameworks lies in the information they provide regarding the model and processing steps used by each of them. In comparison, AutoSklearn offered a lot more information regarding the preprocessing steps and model parameters than AutoGluon. This "extra" information produced by AutoSklearn allows experts to explore preprocessing steps and different model configurations that optimize performance and offer insights about the data and preprocessing steps as well as transparency regarding every step of the AutoML pipeline.

An interesting pattern regarding AutoSklearn observed in our case is that using meta-learning or not did not result in very different performances and it is somewhat visible that when using meta-learning, the model did not improve always for increasing time budgets (see Figure 8). Our hypothesis for this pattern is that AutoSklearn was not specifically designed for RS applications which means that the model pipeline available for meta-learning are not necessarily the best possible, but future research could focus on adapting these AutoML pipelines to RS applications. In terms of data and feature preprocessing and algorithm selection, there was no particular preference. When using meta-learning, we observed more stability in the pipelines being tested (see Figure 9–11) which is expected (Feurer et al. 2015). Nevertheless, in our case, meta-learning did not offer a significant improvement in terms of model fitness. In summary, our results show that AutoML frameworks can be used to improve model performance in the RS and can offer further insights into the preprocessing pipelines.

Conclusions

This study successfully demonstrated the applicability of AutoML for RS of grass height using Landsat ETM and OLI data in the Oostvaardersplassen. Given the limited nature of our training data, we do not consider that our models can be extrapolated to other locations without the inclusion of field data more representative of vegetation height variation. Both AutoML frameworks outperformed an exhaustive feature selection and hyperparameter tuning of commonly used ML algorithms for this task. Our exhaustive exploration of feature combinations (1023 different configurations) on the "classic" approach showed that while ML models benefit from combining bands with vegetation indices, the best performing combination varied per model. This lack of consistency between different models/feature combinations can be addressed by AutoML methods which allowed testing of a wider suite of possibilities and offer top performing result. Furthermore, the AutoML frameworks tested were not designed to address RS challenges or adapted to explore features in the RS domain. There is therefore an open research opportunity in the development of AutoML frameworks specifically for RS applications.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Nuno César de Sá http://orcid.org/0000-0001-7035-5913

Data availability

The authors confirm that the data supporting the findings of this study are available within the article [and/or] its supplementary materials.

References

Ali, I., F. Greifeneder, J. Stamenkovic, M. Neumann, and C. Notarnicola. 2015. "Review of Machine Learning Approaches for Biomass and Soil Moisture Retrievals from Remote Sensing Data." *Remote Sensing* 7 (12): 16398–16421. doi:10.3390/rs71215841.

Balz, T., and F. Rocca. 2020. "Reproducibility and Replicability in SAR Remote Sensing." *IEEE Journal of Selected Topics in*

Applied Earth Observations and Remote Sensing 13: 3834–3843. doi:10.1109/JSTARS.2020.3005912.

Barrachina, M., J. Cristóbal, and A. F. Tulla. 2015. "Estimating Above-Ground Biomass on Mountain Meadows and Pastures through Remote Sensing." *International Journal of Applied Earth Observation and Geoinformation* 38 (June): Elsevier B.V. 184–192. doi: 10.1016/j.jag.2014.12.002.

Belgiu, M., and L. Drăgut. 2016. "Random Forest in Remote Sensing : A Review of Applications and Future Directions." *ISPRS Journal of Photogrammetry and Remote Sensing* 114: 24–31. doi:10.1016/j.isprsjprs.2016.01.011.

Caruana, R., A. Niculescu-Mizil, G. Crew, and A. Ksikes. 2004. "Ensemble Selection from Libraries of Models." In *Twenty-First International Conference on Machine Learning – ICML*, '04 vols. 18. New York, New York, USA: ACM Press. doi:10.1145/1015330.1015432.

Chen, Y., L. Longwei, L. Dengsheng, and L. Dengqiu. 2018. "Exploring Bamboo Forest Aboveground Biomass Estimation Using Sentinel-2 Data." *Remote Sensing* 11 (1): 7. doi:10.3390/rs11010007.

Chen, L., Y. Wang, C. Ren, B. Zhang, and Z. Wang. 2019. "Optimal Combination of Predictors and Algorithms for Forest Above-Ground Biomass Mapping from Sentinel and SRTM Data." *Remote Sensing* 11 (4): 414. doi:10.3390/rs11040414.

Chris, T., F. Hutter, H. H. Hoos, and K. Leyton-Brown. 2013. "Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms." In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 847–855. New York, NY, USA: ACM. doi:10.1145/2487575.2487629.

Colom, M., T. Dagobert, C. de Franchis, R. Grompone von Gioi, C. Hessel, and J.-M. Morel. 2020. "Using the IPOL Journal for Online Reproducible Research in Remote Sensing." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 13: 6384–6390. doi:10.1109/JSTARS.2020.3032100.

Cornelissen, P., M. C. Gresnigt, R. A. Vermeulen, J. Bokdam, and R. Smit. 2014. "Transition of a Sambucus Nigra L. Dominated Woody Vegetation into Grassland by a Multi-Species Herbivore Assemblage." *Journal for Nature Conservation* 22 (1): 84–92. doi:10.1016/j.jnc.2013.09.004.

Craglia, M., J. Hradec, S. Nativi, and M. Santoro. 2017. "Exploring the Depths of the Global Earth Observation System of Systems." *Big Earth Data* 1 (1–2): 21–46. doi:10.1080/20964471.2017.1401284.

Dietterich, T. G. 2000. "Ensemble Methods in Machine Learning." In *Proceedings of the First International Workshop on Multiple Classifier Systems*, 1–15. Verlag: Springer.

Dusseux, P., F. Vertès, T. Corpetti, S. Corgne, and L. Hubert-Moy. 2014. "Agricultural Practices in Grasslands Detected by Spatial Remote Sensing." *Environmental Monitoring and Assessment* 186 (12): 8249–8265. doi:10.1007/s10661-014-4001-5.

Erickson, N., J. Mueller, A. Shirkov, H. Zhang, P. Larroy, M. Li, and A. Smola. 2020. "AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data." *ArXiv Preprint ArXiv:2003.06505*.

Feurer, M., A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter. 2015. "Efficient and Robust Automated Machine Learning." In *Advances in Neural Information Processing Systems 28*, 2962–2970. USA: Curran Associates

Frery, A. C., L. Gomez, and A. C. Medeiros. 2020. "A Badging System for Reproducibility and Replicability in Remote Sensing Research." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 13: 4988–4995. doi:10.1109/JSTARS.2020.3019418.

Gao, L., X. Wang, B. Alan Johnson, Q. Tian, Y. Wang, J. Verrelst, M. Xihan, and G. Xingfa. 2020. "Remote Sensing Algorithms for Estimation of Fractional Vegetation Cover Using Pure Vegetation Index Values: A Review." *ISPRS Journal of Photogrammetry and Remote Sensing* 159 (January): 364–377. doi:10.1016/j.isprsjprs.2019.11.018.

Garroutte, E., A. Hansen, and R. Lawrence. 2016. "Using NDVI and EVI to Map Spatiotemporal Variation in the Biomass and Quality of Forage for Migratory Elk in the Greater Yellowstone Ecosystem." *Remote Sensing* 8 (5): 404. doi:10.3390/rs8050404.

Georganos, S., T. Grippa, S. Vanhuysse, M. Lennert, M. Shimoni, S. Kalogirou, and E. Wolff. 2018. "Less Is More: Optimizing Classification Performance through Feature Selection in a Very-High-Resolution Remote Sensing Object-Based Urban Application." *GIScience & Remote Sensing* 55 (2): 221–242. doi:10.1080/15481603.2017.1408892.

Gorelick, N., M. Hancher, M. Dixon, S. Ilyushchenko, D. Thau, and R. Moore. 2017. "Google Earth Engine: Planetary-Scale Geospatial Analysis for Everyone." *Remote Sensing of Environment* 202 (December): The Author(s). 18–27. doi: 10.1016/j.rse.2017.06.031.

Hanussek, M., M. Blohm, and M. Kintz. 2020. "Can AutoML Outperform Humans? An Evaluation on Popular OpenML Datasets Using AutoML Benchmark." In *2020 2nd International Conference on Artificial Intelligence, Robotics and Control*, 29–32. New York, NY, USA: ACM. doi:10.1145/3448326.3448353.

Huang, S., L. Tang, J. P. Hupy, Y. Wang, and G. Shao. 2021. "A Commentary Review on the Use of Normalized Difference Vegetation Index (NDVI) in the Era of Popular Remote Sensing." *Journal of Forestry Research* 32 (1): 1–6. doi:10.1007/s11676-020-01155-1.

Jingfeng, H., D. Chen, and M. H. Cosh. 2009. "Sub-pixel Reflectance Unmixing in Estimating Vegetation Water Content and Dry Biomass of Corn and Soybeans Cropland Using Normalized Difference Water Index (NDWI) from Satellites." *International Journal of Remote Sensing* 30 (8): 2075–2104. doi:10.1080/01431160802549245.

Kai-Yun, L., R. Sampaio de lima, N. G. Burnside, E. Vahtmäe, T. Kutser, K. Sepp, V. H. Cabral Pinheiro, M.-D. Yang, V. Ants, and K. Sepp. 2022. "Toward Automated Machine Learning-Based Hyperspectral Image Analysis in Crop Yield and Biomass Estimation." *Remote Sensing* 14 (5): 1114. doi:10.3390/rs14051114.

Kim, J.-H. 2009. "Estimating Classification Error Rate: Repeated Cross-Validation, Repeated Hold-out and Bootstrap." *Computational Statistics & Data Analysis* 53 (11): 3735–3745. doi:10.1016/j.csda.2009.04.009.

Koh, J. C. O., G. Spangenberg, and S. Kant. 2021. "Automated Machine Learning for High-Throughput Image-Based Plant

Phenotyping." *Remote Sensing* 13 (5): 858. doi:10.3390/rs13050858.

LeDell, E., and S. Poirier. 2020. "H2O AutoML: Scalable Automatic Machine Learning." *7th ICML Workshop on Automated Machine Learning (AutoML)*, 18th July, 2020, online.

Marcinkowska-Ochtyra, A., A. Jarocińska, K. Bzdęga, and B. Tokarska-Guzik. 2018. "Classification of Expansive Grassland Species in Different Growth Stages Based on Hyperspectral and LiDAR Data." *Remote Sensing* 10 (12): 2019. doi:10.3390/rs10122019.

Marildo, G. F., T. Mora Kuplich, and F. L. F. De Quadros. 2020. "Estimating Natural Grassland Biomass by Vegetation Indices Using Sentinel 2 Remote Sensing Data." In *International Journal of Remote Sensing*, 41 vols (8): 2861–2876. Taylor & Francis. doi:10.1080/01431161.2019.1697004.

Mark J. van der, L., E. C. Polley, and A. E. Hubbard. 2007. "Super Learner." *Statistical Applications in Genetics and Molecular Biology* 6: 1. doi:10.2202/1544-6115.1309.

Maxwell, A. E., T. A. Warner, and F. Fang. 2018. "Implementation of Machine-Learning Classification in Remote Sensing: An Applied Review." *International Journal of Remote Sensing* 39 (9): 2784–2817. doi:10.1080/01431161.2018.1433343.

Murphy, S. W., C. R. de Souza Filho, R. Wright, G. Sabatino, and R. Correa Pabon. 2016. "HOTMAP: Global Hot Target Detection at Moderate Spatial Resolution." *Remote Sensing of Environment* 177 (May): 78–88. doi:10.1016/j.rse.2016.02.027.

Mutanga, O., E. Adam, and M. Azong Cho. 2012. "High Density Biomass Estimation for Wetland Vegetation Using WorldView-2 Imagery and Random Forest Regression Algorithm." *International Journal of Applied Earth Observation and Geoinformation* 18 (1): 399–406. Elsevier B. V. doi:10.1016/j.jag.2012.03.012.

Olson, R. S., R. J. Urbanowicz, P. C. Andrews, N. A. Lavender, L. Creis Kidd, and J. H. Moore. 2016. "Automating Biomedical Data Science Through Tree-Based Pipeline Optimization." *In* 123–137. doi:10.1007/978-3-319-31204-0_9.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. "Scikit-Learn: Machine Learning in Python." *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 12 vols. 2825–2830, Prague, 23-27 September, 2013.

Peng, C., L. Yangyang, L. Jiao, and R. Shang. 2021. "Efficient Convolutional Neural Architecture Search for Remote Sensing Image Scene Classification." *IEEE Transactions on Geoscience and Remote Sensing* 59 (7): 6092–6105. doi:10.1109/TGRS.2020.3020424.

Pettorelli, N., H. S. to Bühne, A. Tulloch, G. Dubois, C. Macinnis-Ng, A. M. Queirós, D. A. Keith, et al. 2018. "Satellite Remote Sensing of Ecosystem Functions: Opportunities, Challenges and Way Forward." In *Remote Sensing in Ecology and Conservation*, Edited by M. Rowcliffe and M. Disney, 4 vols (2): 71–93. Hoboken, New Jersey: John Wiley & Sons, Ltd. doi:10.1002/rse2.59.

Pham, T. D., L. Nga Nhu, H. Nam Thang, L. Viet Nguyen, J. Xia, N. Yokoya, T. Tu Trong, H. Xuan Trinh, L. Quoc Kieu, and W. Takeuchi. 2020. "Estimating Mangrove Above-Ground Biomass Using Extreme Gradient Boosting Decision Trees Algorithm with Fused Sentinel-2 and ALOS-2 PALSAR-2 Data in Can Gio Biosphere Reserve, Vietnam." *Remote Sensing* 12 (5): 777. doi:10.3390/rs12050777.

Phiri, D., M. Simwanda, S. Salekin, V. R. Nyirenda, Y. Murayama, and M. Ranagalage. 2020. "Sentinel-2 Data for Land Cover/Use Mapping: A Review." *Remote Sensing* 12 (14): 2291. doi:10.3390/rs12142291.

Pôças, I., A. Calera, I. Campos, and M. Cunha. 2020. "Remote Sensing for Estimating and Mapping Single and Basal Crop Coefficients: A Review on Spectral Vegetation Indices Approaches." *Agricultural Water Management* 233 (April): 106081. doi:10.1016/j.agwat.2020.106081.

Reinermann, S., S. Asam, and C. Kuenzer. 2020. "Remote Sensing of Grassland Production and Management—A Review." *Remote Sensing* 12 (12): 1949. doi:10.3390/rs12121949.

Reza, K., G. Mountrakis, and S. V. Stehman. 2016. "A Meta-Analysis of Remote Sensing Research on Supervised Pixel-Based Land-Cover Image Classification Processes: General Guidelines for Practitioners and Future Research." *Remote Sensing of Environment* 177 (May): 89–100. doi:10.1016/j.rse.2016.02.028.

Sahar, D., S. L. Cutter, and C. Wang. 2020. "Remote Sensing Derived Indices for Tracking Urban Land Surface Change in case of Earthquake Recovery." *Remote Sensing* 12 (5): 895. doi:10.3390/rs12050895.

Salinas, N. R. P., M. Baratchi, J. N. van Rijn, and A. Vollrath. 2021. "Automated Machine Learning for Satellite Data: Integrating Remote Sensing Pre-Trained Models into AutoML Systems." In *Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track*, edited by Y. Dong, N. Kourtellis, B. Hammer, and J. A. Lozano, 447–462. Cham: Springer International Publishing.

Schulze-Brüninghoff, D., M. Wachendorf, and T. Astor. September 2020. "Remote Sensing Data Fusion as a Tool for Biomass Prediction in Extensive Grasslands Invaded by L. Polyphyllus." In *Remote Sensing in Ecology and Conservation*, edited by M. Disney and S. Levick, rse2.182. Hoboken, New Jersey: John Wiley & Sons, Ltd. doi:10.1002/rse2.182

Shih, H.-C., D. A. Stow, and Y. Hsin Tsai. 2019. "Guidance on and Comparison of Machine Learning Classifiers for Landsat-Based Land Cover and Land Use Mapping." *International Journal of Remote Sensing* 40 (4): 1248–1274. doi:10.1080/01431161.2018.1524179.

Staatsbosbeheer. 2021. "Oostvaaadersplassen." https://www.staatsbosbeheer.nl/uit-in-de-natuur/locaties/oostvaardersplassen

Stine, R. A. 1995. "Graphical Interpretation of Variance Inflation Factors." In *The American Statistician*. 49 vols (1): 53–56. Milton Park, Oxfordshire: Taylor & Francis. doi: 10.1080/00031305.1995.10476113.

Stromann, O., A. Nascetti, O. Yousif, and Y. Ban. 2019. "Dimensionality Reduction and Feature Selection for Object-Based Land Cover Classification Based on Sentinel-1 and Sentinel-2 Time Series Using Google Earth Engine." *Remote Sensing* 12 (1): 76. doi:10.3390/rs12010076.

Tang, Y., A. Ali, and L.-H. Feng. 2020 "Bayesian Model Predicts the Aboveground Biomass of Caragana Microphylla in Sandy Lands Better than OLS Regression Models." In *Journal of*

Plant Ecology, edited by E. Bai, 13 vols (6): 732–737. UK: Oxford University Press. doi:10.1093/jpe/rtaa065

Taravat, A., M. Wagner, and N. Oppelt. 2019. "Automatic Grassland Cutting Status Detection in the Context of Spatiotemporal Sentinel-1 Imagery Analysis and Artificial Neural Networks." *Remote Sensing* 11 (6): 711. doi:10.3390/rs11060711.

Tedesco, D., M. Freire de Oliveira, A. Felipe Dos Santos, E. Henrique Costa Silva, G. de Souza Rolim, and R. Pereira da Silva. 2021. "Use of Remote Sensing to Characterize the Phenological Development and to Predict Sweet Potato Yield in Two Growing Seasons." *European Journal of Agronomy* 129 (September): 126337. doi:10.1016/j.eja.2021.126337.

Ullah, S., S. Yali, M. Schlerf, A. K. Skidmore, M. Shafique, and I. Akhtar Iqbal. 2012. "Estimation of Grassland Biomass and Nitrogen Using MERIS Data." *International Journal of Applied Earth Observation and Geoinformation* 19 (October): 196–204. doi:10.1016/j.jag.2012.05.008.

Vafaei, S., J. Soosani, K. Adeli, H. Fadaei, H. Naghavi, T. Pham, and D. Tien Bui. 2018. "Improving Accuracy Estimation of Forest Aboveground Biomass Based on Incorporation of ALOS-2 PALSAR-2 and Sentinel-2A Imagery and Machine Learning: A Case Study of the Hyrcanian Forest Area (Iran)." *Remote Sensing* 10 (2): 172. doi:10.3390/rs10020172.

Wang, G. 2019. "Machine Learning for Inferring Animal Behavior from Location and Movement Data." *Ecological Informatics* 49 (January): 69–76. doi:10.1016/j.ecoinf.2018.12.002.

Wang, C., T. Back, H. H. Hoos, M. Baratchi, S. Limmer, and M. Olhofer. 2019a. "Automated Machine Learning for Short-Term Electric Load Forecasting." In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, Xiamen, China, 6-9 December, 2019, 314–321. IEEE. doi:10.1109/SSCI44817.2019.9002839.

Wang, Z., P. A. Townsend, A. K. Schweiger, J. J. Couture, A. Singh, S. E. Hobbie, and J. Cavender-Bares. 2019b. "Mapping Foliar Functional Traits and Their Uncertainties across Three Years in a Grassland Experiment." *Remote Sensing of Environment* 221 (February): 405–416. doi:10.1016/j.rse.2018.11.016.

Wang, J., X. Xiao, R. Bajgain, P. Starks, J. Steiner, R. B. Doughty, and Q. Chang. 2019. "Estimating Leaf Area Index and Aboveground Biomass of Grazing Pastures Using Sentinel-1, Sentinel-2 and Landsat Images." *ISPRS Journal of Photogrammetry and Remote Sensing* 154 (June): Elsevier. 189–201. doi: 10.1016/j.isprsjprs.2019.06.007.

Wang, J., Y. Zhong, Z. Zheng, M. Ailong, and L. Zhang. 2021. "RSNet: The Search for Remote Sensing Deep Neural Networks in Recognition Tasks." *IEEE Transactions on Geoscience and Remote Sensing* 59 (3): 2520–2534. doi:10.1109/TGRS.2020.3001401.

Wang, L., X. Zhou, X. Zhu, Z. Dong, and W. Guo. 2016 "Estimation of Biomass in Wheat Using Random Forest Regression Algorithm and Remote Sensing Data." *The Crop Journal*. 4 vols (3): 212–219. Amsterdam, the Netherlands: Elsevier B.V. doi: 10.1016/j.cj.2016.01.008.

Wen, W., J. Timmermans, Q. Chen, and P. M. van Bodegom. 2020. "A Review of Remote Sensing Challenges for Food Security with respect to Salinity and Drought Threats." *Remote Sensing* 13 (1): 6. doi:10.3390/rs13010006.

Wijesingha, J., T. Astor, D. Schulze-Brüninghoff, M. Wengert, and M. Wachendorf. 2020. "Predicting Forage Quality of Grasslands Using UAV-Borne Imaging Spectroscopy." *Remote Sensing* 12 (1): 126. doi:10.3390/rs12010126.

Wilkinson, M. D., M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg. et al. 2016. "The FAIR Guiding Principles for Scientific DataScientific Data Management and Stewardship." *Scientific DataScientific Data* 3 (1): 160018. doi:10.1038/sdata.2016.18.

Xie, L., L. Fengri, L. Zhang, F. Rafi Almay Widagdo, and L. Dong. 2020. "A Bayesian Approach to Estimating Seemingly Unrelated Regression for Tree Biomass Model Systems." *Forests* 11 (12): 1–30. doi:10.3390/f11121302.

Xie, Y., Z. Sha, Y. Mei, Y. Bai, and L. Zhang. 2009. "A Comparison of Two Models with Landsat Data for Estimating above Ground Grassland Biomass in Inner Mongolia, China." *Ecological Modelling* 220 (15): 1810–1818. doi:10.1016/j.ecolmodel.2009.04.025.

Xin, H., K. Zhao, and X. Chu. 2021. "AutoML: A Survey of the State-of-the-Art." *Knowledge-Based Systems* 212 (January): 106622. doi:10.1016/j.knosys.2020.106622.

Yang, S., Q. Feng, T. Liang, B. Liu, W. Zhang, and H. Xie. 2018. "Modeling Grassland Above-Ground Biomass Based on Artificial Neural Network and Remote Sensing in the Three-River Headwaters Region." *Remote Sensing of Environment* 204: 448–455. doi: 10.1016/j.rse.2017.10.011. March 2017. Elsevier.

Yang, L., and A. Shami. 2020. "On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice." *Neurocomputing* 415 (November): Elsevier B.V. 295–316. doi: 10.1016/j.neucom.2020.07.061.

Yongjie, J., X. Kunpeng, P. Zeng, and W. Zhang. 2021. "GA-SVR Algorithm for Improving Forest above Ground Biomass Estimation Using SAR Data." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14: 6585–6595. doi:10.1109/JSTARS.2021.3089151.

Zhang, M., W. Jing, J. Lin, N. Fang, W. Wei, M. Woźniak, and R. Damaševičius. 2020a. "NAS-HRIS: Automatic Design and Architecture Search of Neural Network for Semantic Segmentation in Remote Sensing Images." *Sensors* 20 (18): 5292. doi:10.3390/s20185292.

Zhang, Y., M. Jun, S. Liang, L. Xisheng, and L. Manyao. 2020b. "An Evaluation of Eight Machine Learning Regression Algorithms for Forest Aboveground Biomass Estimation from Multiple Satellite Data Products." *Remote Sensing* 12 (24): 4015. doi:10.3390/rs12244015.

Zhu, J., Z. Huang, H. Sun, and G. Wang. 2017. "Mapping Forest Ecosystem Biomass Density for Xiangjiang River Basin by Combining Plot and Remote Sensing Data and Comparing Spatial Extrapolation Methods." *Remote Sensing* 9 (3): 241. doi:10.3390/rs9030241.

Zhuo, W., J. Huang, X. Zhang, H. Sun, D. Zhu, S. Wei, C. Zhang, and Z. Liu. 2016. "Comparison of Five Drought Indices for Agricultural Drought Monitoring and Impacts on Winter Wheat Yields Analysis." *2016 5th International Conference on Agro-Geoinformatics, Agro-Geoinformatics 2016,* Tianjin, China, 18-20 July. doi:10.1109/Agro-Geoinformatics.2016.7577702.

Appendix 1: Landsat data description

Sensor	Raw number	Common denomination	Wavelength (μm)	Resolution (m)	New band number
ETM+	**1**	**Blue**	**0.45–0.52**	**30**	**1**
	2	**Green**	**0.52–0.60**	**30**	**2**
	3	**Red**	**0.63–0.69**	**30**	**3**
	4	**NIR**	**0.77–0.90**	**30**	**4**
	5	**SWIR**	**1.55–1.75**	**30**	**5**
	6	Thermal	10.40–12.50	60	-
	7	**Mid-infrared**	**2.09–2.35**	**30**	**7**
	8	Panchromatic	0.52–0.90	15	-
OLI	1	Coastal aerosol	0.43–0.45	30	-
	2	**Blue**	**0.45–0.51**	**30**	**1**
	3	**Green**	**0.53–0.59**	**30**	**2**
	4	**Red**	**0.64–0.67**	**30**	**3**
	5	**NIR**	**0.87–0.88**	**30**	**4**
	6	**SWIR 1**	**1.57–1.65**	**30**	**5**
	7	**SWIR 2**	**2.11–2.29**	**30**	**7**
	8	Panchromatic	0.50–0.68	15	-
	9	Cirrus	1.36–1.38	30	-
	10	TIRS 1	10.6–11.19	100	-
	11	TIRS 2	11.50–12 – 51	100	-

The above table shows the bands used. Bold shows the selected bands on each sensor andthe new band number attributed to facilitate comparison of the results from both sensors.

Appendix 2: Parameter spaces of MLR hyperparameter optimization

Algorithm	Description	Parameter	Space	References
Support vector machine	Kernel trick functions	kernel	Poly, rbf, sigmoid	(Shih, Stow, and Hsin Tsai 2019; Wang et al. 2016; Yang and Shami 2020)
	Kernel coefficient	gamma	auto	
	Degree of polynomial kernel	degree	Int(1,3)	
	Penalty associated with training loss	epsilon	Real (0.0001,100)	
	Regularization parameter	C	Real (0.01,10,000)	
	Limit on solver iterations	max_iter	Int(100,1000)	
Random forest	Number of trees	n_estimators	Int(50,1000)	(Belgiu & Drăguț, 2016; Mutanga, Adam, and Azong Cho 2012; Shih, Stow, and Hsin Tsai 2019; Wang et al. 2016; Yang and Shami 2020)
	Maximum depth of a tree	max_depth	Int(1,50)	
	Minimum samples on internal node split	min_samples_split	Real(0,0.50)	
	Minimum samples required to be a leaf node	min_samples_leaf	Real(0,0.50)	
	Maximum samples drawn to train base models	max_samples	0.5	
K-nearest neighbor	Number of neighboring samples	n_neighbors	int(5,50)	(Yang and Shami 2020; Zhu et al. 2017)
	Leaf size parameter passed to nn algorithm	leaf_size	Int(2,50)	
	Weighting function for sample distance	weights	uniform, distnace	
Bayesian ridge regression	Shape parameter of the prior Gamma distribution over the alpha parameter	alpha_1	Real(1e^{-7},1)	(Tang, Ali, and Feng 2020; Xie et al. 2020; Yang and Shami 2020)
	Inverse scale parameter for the prior Gamma distribution over the alpha parameter	alpha_2	Real(1e^{-7},1)	
	Shape parameter for the Gamma distribution over the lambda parameter	lambda_1	Real(1e^{-7},1)	
	Inverse scale parameter for the prior Gamma distribution over the Lambda parameter	lambda_2	Real(1e^{-7},1)	
Artificial neural networks	Number of neurones on hidden layer	hidden_layer_sizes	Int(1,30)	(Taravat, Wagner, and Oppelt 2019; Wang et al. 2016; Yang and Shami 2020)
	Type of neuron activation function	activation	identity, logistic, tanh, relu	
	Adaptive or non-adaptative learning rate	learning_rate	constant, invscaling, adaptive	

The above table summarizes the sets of hyperparameters that were tuned for each of the“classic” machine learning models, the intervals and type of each parameter is alsoshown. The bibliographic references used to select these intervals is shown on therightmost column.

Appendix 3

Column name	Description
Sampling Date	Date of field data sampling
TransectID	ID of transect
Distance	Distance from transect start
grass_height	Height field estimate
grass_height + error	Height measurement error
RS image date	Date of RS imagery acquisition
diffDay	Difference between RS image and field sample (in days)
SensorType	Sensor
PathRow	Path/row of Landsat imagery
X_coord	X coordinate of sample
Y_coord	Y coordinate of sample#
B1	Reflectance for each band
B2	
B3	
B4	
B5	
B7	
QA_Band	Quality assessment value
NDVI	Normalized difference vegetation index
EVI	Enhanced normalized vegetation index
SAVI	Soil adjusted vegetation index
NDWI	Normalized difference water index

Appendix 4

Appendix_03	Description
Model	Model
Features	Features used for the model
Validation type	Validation type
MAE	Mean absolute error
MAEsd	Standard deviation of MAE
In case of AutoML	
TimeBudget	Time in seconds used for model optimization