International Journal of Applied Earth Observations and Geoinformation 106 (2022) 102657

Contents lists available at ScienceDirect

International Journal of Applied Earth Observations and Geoinformation

journal homepage: www.elsevier.com/locate/jag

Multi-target regressor chains with repetitive permutation scheme for characterization of built environments with remote sensing

Christian Geiß [a,*], Elisabeth Brzoska [a], Patrick Aravena Pelizari [a], Sven Lautenbach [b], Hannes Taubenböck [a]

[a] German Remote Sensing Data Center (DFD), German Aerospace Center (DLR), 82234 Weßling-Oberpfaffenhofen, Germany
[b] Heidelberg University, GIScience Research Group, 69120 Heidelberg, Germany

ARTICLE INFO

Keywords:
Multi-target regression
Single-target stacking
Multi-target stacking
Regressor chains
Built-up density
Built-up height

ABSTRACT

Multi-task learning techniques allow the beneficial joint estimation of multiple target variables. Here, we propose a novel multi-task regression (MTR) method called ensemble of regressor chains with repetitive permutation scheme. It belongs to the family of problem transformation-based MTR methods which foresee the creation of an individual model per target variable. Subsequently, the combination of the separate models allows obtaining an overall prediction. Our method builds upon the concept of so-called ensemble of regressor chains which align single-target models along a flexible permutation, i.e., chain. However, in order to particularly address situations with a small number of target variables, we equip ensemble of regressor chains with a repetitive permutation scheme. Thereby, estimates of the target variables are cascaded to subsequent models as additional features when learning along a chain, whereby one target variable can occupy multiple elements of the chain. We provide experimental evaluation of the method by jointly estimating *built-up height* and *built-up density* based on features derived from Sentinel-2 data for the four largest cities in Germany in a comparative setup. We also consider single-target stacking, multi-target stacking, and ensemble of regressor chains without repetitive permutation. Empirical results underline the beneficial performance properties of MTR methods. Our ensemble of regressor chain with repetitive permutation scheme approach achieved most frequently the highest accuracies compared to the other MTR methods, whereby mean improvements across the experiments of 14.5% compared to initial single-target models could be achieved.

1. Introduction

The automated extraction of thematic information from remote sensing (RS) data is frequently addressed by resorting to supervised learning techniques. Such techniques foresee assigning a thematic label to an instance given a sufficient amount of properly encoded prior knowledge, i.e., training data. The training data is deployed to infer a rule, e.g., a decision function, which aims to generalize well for unseen instances (Geiß et al., 2019a).

In the context of RS, typical supervised models assign a single thematic label such as a land use/land cover (LULC) class to an instance (e.g., an image pixel or object). It was shown that algorithms such as Support Vector Machines (Cortes & Vapnik, 1995; Camps-Valls & Bruzzone, 2005), Random Forest (RF) (Breimann, 2001; Pal, 2005), and Convolutional Neural Networks (CNN) (LeCun et al., 2015), among others, are able to efficiently address this task.

Particular applications in RS require the assignment of multiple thematic labels to an instance (Geiß et al., 2017a). Such labels can be of discrete or continuous nature. Especially classification-related tasks (i.e., predicting multiple categorical labels simultaneously) have gained much attention in the past. Tailored techniques to address this task are called multi-label classification (MLC) methods (Tsoumakas & Katakis, 2007; Read et al., 2011). However, in this paper we are interested in the case when the labels of the target variables are continuous. Consequently, we consider multi-target regression (MTR) methods (Borchani et al., 2015). The governing principle of MTR is to model the intrinsic correlation of the target variables, which can largely improve parameter estimation by sharing knowledge across correlated outputs.

While MTR received increasing attention in the machine learning community in recent years (Li et al, 2021), only few authors have

[*] Corresponding author.
E-mail addresses: christian.geiss@dlr.de (C. Geiß), elisabeth.brzoska@dlr.de (E. Brzoska), patrick.aravenapelizari@dlr.de (P. Aravena Pelizari), sven.lautenbach@heigit.org (S. Lautenbach), hannes.taubenboeck@dlr.de (H. Taubenböck).

https://doi.org/10.1016/j.jag.2021.102657
Received 23 February 2021; Received in revised form 1 June 2021; Accepted 15 December 2021
Available online 27 December 2021

focused on these methods within the RS domain to date: Kocev et al. (2009) deploy MTR trees (MTRTs) and ensembles of MTRTs to predict seven measures which characterize the condition of vegetation based on Landsat 7 data. Stojanova et al. (2010) generate training data from LiDAR measurements to estimate vegetation height and canopy cover with wide geospatial coverage also with MTRTs/ensembles of MTRTs and Landsat 7 data. Tuia et al. (2011) alter the cost function of a Support Vector Regression (SVR) model to simultaneously predict chlorophyll content, leaf area index, and fractional vegetation cover using hyperspectral imagery. Subsequently, Rosentreter et al. (2017) deploy this MTR SVR model for subpixel analysis on a simulated environmental mapping and analysis program (EnMAP) scene. Hence, land cover fraction maps are derived for an extended so-called V-I-S mapping scheme, consisting of the target variables vegetation, impervious surface, soil, and water. Recently, Mandal et al. (2019) estimate plant area index (PAI) and wet biomass of soybean and wheat using a multi-target RF regression model learned on full-polarimetric RADARSAT-2 data. Dey, 2021 propose an encode-decoder regression network for simultaneous estimation of PAI and wet biomass from SAR data. Pyo et al. (2019) establish an MTR CNN model with two branches for joint estimation of concentrations of phycocyanin and chlorophyll-a based on hyperspectral imagery. The mentioned studies report beneficial generalization capabilities of the MTR models with respect to individual models and achievement of less dependent residuals, while maintaining higher efficiency regarding costs for learning and inference.

From a model perspective, a dichotomy between *problem transformation methods* and *algorithm adaptation* methods can be drawn (Tsoumakas & Katakis, 2007; Borchani et al., 2015). The aforementioned RS-based studies all utilize *algorithm adaptation* methods. Algorithm adaptation methods adapt a specific single-output method to directly handle multi-output problems and predict all targets at once by, e.g., imposing regularization terms on the regression weights to explore output relationships. It was argued, that algorithm adaptation methods are easier to interpret and more scalable to large output spaces compared to numerous problem transformation methods, as they only establish a single multi-target model. In this manner, dedicated MTR extensions for advanced regression algorithms such as SVR (Li et al., 2020b; Sanchez-Fernandez et al., 2004), Decision Trees (Segal and Xiao, 2011), Gaussian Process Regression (Liu et al., 2018), Neural Networks (Reyes and Ventura, 2019), and Outlier Extreme Learning Machine (Souza da Silva, 2020), among others, were proposed.

However, in this paper we are interested in *problem transformation* methods. Such techniques foresee the creation of an individual model per target, and then the combination of the separate models in order to obtain an overall prediction. Problem transformation methods have been found superior to algorithm adaptation methods in terms of accuracy (Spyromitros-Xioufis et al., 2016). Moreover, the underlying principle renders problem transformation methods independent of the regression algorithm. Consequently, they can be easily adapted to the problem at hand by employing suitable base learners. This point is also of particular relevance for ensemble models which concatenate the estimates from multiple possibly different regression algorithms into a final prediction. Such techniques were found to be beneficial in terms of accuracy and reliability properties of model estimates in the context of RS (Geiß et al., 2020; Feilhauer et al., 2015).

A fundamental concept in *problem transformation* methods is to utilize preceding models for a new prediction via an extended feature space (Borchani et al., 2015). This concept is related to stacked generalization which was introduced by Wolpert (1992). Stacked generalization is a *meta*-learning approach which deploys the outputs of previously learned models for learning a new model. As such, the initial model outputs are treated as new features and are stacked to the initial feature vector before relearning. In the original formulation only a two-stage procedure was foreseen, i.e., the initial models learned from the initial feature vector correspond to level–0 models and data, respectively, and the enlarged feature vector and the relearned model are referred to as level–1 data and generalizer, respectively (Wolpert, 1992). However, reasonably, this single-target stacking (STS) process can also be carried out over multiple iterations. In order to deploy this principle for multi-target problems, where also possible correlations among the target variables are encoded, the idea of multi-target stacking (MTS) was introduced (Borchani et al., 2015; Spyromitros-Xioufis et al., 2016). Analogous to STS, training an MTS model can be regarded as a two-stage procedure. In the first stage, independent models for each target variable are learned. Subsequent to this, *meta*-models for each target variable are learned with expanded feature vectors, which contain the initial feature vectors and also the level–0 estimates of the residual target variables (Spyromitros-Xioufis et al., 2016). Likewise ideas were also followed in the context of ensemble models, i.e., learning several level–0 models for each target variable which are concatenated in a level–1 generalization procedure for multiple target variables (Santana et al., 2020).

As an alternative strategy for MTR, the concept of regressor chains (RC) was introduced (Spyromitros-Xioufis et al., 2016). In the most basic formulation, RC foresee the linkage of single-target models according to a chain structure. First, a random chain (i.e., permutation) is drawn based on the set of target variables. Then, separate regression models are established for each target variable while following the order of the chain. Such a strategy was found beneficial in terms of accuracy properties compared to other *problem transformation-based* MTR approaches (Spyromitros-Xioufis et al., 2016). At the same time RC were found to be sensitive to the ordering of the chain. Consequently, ensemble of RC were proposed where a set of regression chains with differing order of the chain elements is learned (Spyromitros-Xioufis et al., 2016). Subsequent works also aimed to optimize the ordering of the chain, e.g., by employing a correlation measure with respect to the target variables (Melki et al., 2017).

Thereby, existing approaches foresee solely the singular occurrence of a target variable in a chain. Spyromitros-Xioufis et al. (2016) propose ensemble of RC where the number of elements in the chain is governed by the number of target variables. An ensemble of RC is established where each chain with length m consists of distinct permutations of the target variables. However, to keep computational complexity low, the number of chains and elements of a chain, respectively, is generally constrained to ten.

Such a strategy discards the possibility to build chains which benefit from possibly enhanced estimates of the target variables while sequentially learning along the chain, especially when the number of target variables is small. Consequently, in contrast to existing approaches, we equip ensemble of RC with a repetitive permutation scheme with respect to the elements of the chain. Thus, chains with a larger number of elements than the number of target variables can be established. This generates high level transformations of the original inputs and can lead to an improved performance of the chaining model (Read and Hollmen, 2015). As such, we consider the main contributions of this paper as follows.

(1) From a methodological point of view, we propose multi-target regressor chains with repetitive permutation scheme, where estimates of the target variables are cascaded to subsequent models as additional features when learning along the chain. Thereby, chains with a larger number of elements than the number of target variables are established, i.e., one target variable can occupy multiple elements of the chain.

(2) We carry out a systematic analysis where we decompose the influence of the subcomponents of the method on the prediction accuracy. This analysis comprises the systematic disentanglement with respect to model accuracy according to the considered MTR method (i.e., STS, MTS, ensemble of RC with and without repetitive permutation), regression algorithm (i.e., Random Forest Regression, Support Vector Regression, Gaussian Process Regression, and Neural Network Regression are evaluated), augmentation strategy for the feature vector (i.e., including

previous model predictions in a cumulative and non-cumulative way), and composition of the feature space (i.e., deploying target variable-specific features or a set of features which is shared over all target variables for prediction).

(3) We provide an exhaustive experimental evaluation of the methods in the context of an innovative application domain. Multi-target models are deployed to simultaneously estimate *built-up density* and *height* based on Sentinel-2 data. Recently, efforts were carried out to derive constituting properties of the urban morphology for large areas based on supervised learning techniques using Sentinel data (Geiß et al., 2020; Frantz et al., 2021). In this context, we learn models from training data sets created with Level of Detail 1 (LoD-1) building models, whereby buildings are represented by extruded footprints, and jointly regress the two target variables for spatial processing units which correspond to urban neighborhood scales for the four largest German cities (i.e., Berlin, Hamburg, Munich, and Cologne) using features calculated from ubiquitously available multispectral Sentinel-2 imagery.

The remainder of the paper is organized as follows. Section 2 gives an overview of the MTR methods and Section 3 is used to present the deployed data sets and explain the experimental setup. Section 4 provides experimental results and validation efforts. Concluding remarks are given in Section 5.

2. Multi-task regression methods

Let us consider a training data set $D = \left\{ \left(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}\right), \cdots, \left(\mathbf{x}^{(N)}, \mathbf{y}^{(N)}\right) \right\}$ composed of N instances, i.e., labeled samples, with a value assignment for each variable $X_1, \cdots, X_m, Y_1, \cdots, Y_d$. Thus, each instance is represented by an input vector of m predictive variables $\mathbf{x}^{(l)} = \left(x_1^{(l)}, \cdots, x_j^{(l)}, \cdots, x_m^{(l)}\right)$ and corresponding output vector with multiple d target variables $\mathbf{y}^{(l)} = \left(y_1^{(l)}, \cdots, y_i^{(l)}, \cdots, y_d^{(l)}\right)$, with $l \in \{1, \cdots, N\}, j \in \{1, \cdots, m\}$, and $i \in \{1, \cdots, d\}$. In MTR, the goal is to draw a hypothesis h from D, which predicts an output vector $\mathbf{y}$ of d target values to an instance, given an input vector $\mathbf{x}$:

$$h : \Omega_{X_1} \times \cdots \times \Omega_{X_m} \rightarrow \Omega_{Y_1} \times \cdots \times \Omega_{Y_d} \tag{1}$$

$$\mathbf{x} = (x_1, \cdots, x_m) \mapsto \mathbf{y} = (y_1, \cdots, y_d), \tag{2}$$

where Ω_{X_j} and Ω_{Y_i} represent the sample spaces of an individual predictive variable X_j, for all $j \in \{1, \cdots, m\}$, and an individual target variable Y_i, for all $i \in \{1, \cdots, d\}$, respectively (Borchani et al., 2015).

2.1. Single-target stacking

STS predicts all target variables independently with respect to each other. Nevertheless, an MTR STS model comprises d single-target models, whereby the value of an individual target variable Y_i is predicted. After initial training, a *meta*-model is learned from a transformed training data set $D_i^{'}$ whereby the initial training data set D_i was augmented by previous model estimates with respect to Y_i, i.e., $D_i^{'} = \left\{ \left(\mathbf{x}^{'(1)}, y_i^{(1)}\right), \cdots, \left(\mathbf{x}^{'(N)}, y_i^{(N)}\right) \right\}$, where $\mathbf{x}^{'(l)} = \left(x_1^{(l)}, \cdots, x_N^{(l)}, \widehat{y}_i^{(l)}\right)$ (Fig. 1a). We implement the case where the transformed training data set is compiled in a cumulative and non-cumulative way when training the models over multiple iterations. A cumulative augmentation strategy for the feature vector foresees the inclusion of the estimates of all preceding models, i.e., the training data set is constructed as follows after the second iteration: $\overline{\overline{D}}_i^{''} = \left\{ \left(\mathbf{x}^{''(1)}, y_i^{(1)}\right), \cdots, \left(\mathbf{x}^{''(N)}, y_i^{(N)}\right) \right\}$, where $\mathbf{x}^{''(l)} = \left(x_1^{(l)}, \cdots, x_N^{(l)}, \widehat{y}_i^{(l)}, \widehat{y}_i^{'(l)}\right)$. In contrast, a non-cumulative augmentation strategy for the feature vector foresees solely the inclusion of the estimates of the preceding model, i.e., the training data set is constructed as follows after the second iteration: $\overline{D}_i^{''} = \left\{ \left(\mathbf{x}^{''(1)}, y_i^{(1)}\right), \cdots, \left(\mathbf{x}^{''(N)}, y_i^{(N)}\right) \right\}$, where $\mathbf{x}^{''(l)} = \left(x_1^{(l)}, \cdots, x_N^{(l)}, \widehat{y}_i^{'(l)}\right)$. Consequently, after the second iteration, the dimensionality of the training data set remains constant for the non-cumulative procedure. In contrast, the dimensionality of the training data set linearly increases in the cumulative case as a function of the number of iterations. In the context of so-called relearning procedures, which compute new features in the geospatial domain based on preliminary model estimates, it was shown that both a non-cumulative and cumulative augmentation of the feature vector can yield competitive predictions (Geiß, et al., in press). We implement both strategies for all considered MTR methods and evaluate their potential effect on the prediction accuracy later on.

The training set is altered after each iteration where Y_i remains the same, while the feature vector is transformed and extended for the non-cumulative and the cumulative case, respectively. Thereby, the number of iterations can be set according to a stopping criterion. Finally, as it is done with all MTR methods in this work, the model which maximizes a defined accuracy measure among all learned models is selected. As such, we treat the number of iterations as a hyperparameter which needs to be optimized in a data-driven way without prior constraints.

2.2. Multi-target stacking

In contrast to STS, MTS is designed to actually share knowledge across correlated target variables within a stacking procedure. Likewise, d single-target models are learned first. Subsequently, a set of *meta*-models is established which include a model for each target variable Y_i, $i \in \{1, \cdots, d\}$. Thereby, estimates regarding the residual target variables from the first stage are included, i.e., a model is learned from a transformed data set $D_i^{'} = \left\{ \left(\mathbf{x}^{'(1)}, y_i^{(1)}\right), \cdots, \left(\mathbf{x}^{'(N)}, y_i^{(N)}\right) \right\}$, where $\mathbf{x}^{'(l)} = \left(x_1^{(l)}, \cdots, x_N^{(l)}, \widehat{y}_1^{(l)}, \cdots, \widehat{y}_d^{(l)}\right)$ (Fig. 1b). Also here, we distinguish the cumulative and non-cumulative training strategy after the second iteration where $\overline{\overline{D}}_i^{''}$ and $\overline{D}_i^{''}$ contain $\mathbf{x}^{''(l)} = \left(x_1^{(l)}, \cdots, x_N^{(l)}, \widehat{y}_1^{(l)}, \cdots, \widehat{y}_d^{(l)}, \widehat{y}_1^{'(l)}, \cdots, \widehat{y}_d^{'(l)}\right)$ and $\mathbf{x}^{''(l)} = \left(x_1^{(l)}, \cdots, x_N^{(l)}, \widehat{y}_1^{'(l)}, \cdots, \widehat{y}_d^{'(l)}\right)$, respectively.

In order to be able to obtain predictions with stacking methods for an unseen instance, i.e., unlabeled sample $\mathbf{x}^{(N+1)}$, estimates based on initial models are generated first, i.e., inducing the estimated output vector $\widehat{\mathbf{y}}^{(N+1)} = \left(\widehat{y}_1^{(N+1)}, \cdots, \widehat{y}_d^{(N+1)}\right)$. Then, the *meta*-models are applied on the transformed input vector $\mathbf{x}^{'(N+1)} = \left(x_1^{(N+1)}, \cdots, x_m^{(N+1)}, \widehat{y}_1^{(N+1)}, \cdots, \widehat{y}_d^{(N+1)}\right)$ to produce new predictions for the target variables.

2.3. Regressor chains

RC are based on the idea of aligning single-target models along a flexible permutation, i.e., chain. First, a permutation with respect to the target variables is drawn. This process can be carried out in a random (e.g., Spyromitros-Xioufis et al., 2016) or directed (e.g., Melki et al., 2017) manner. The selected permutation is deployed to build a separate regression model for the target variables following the order of the permutation. To exploit this structure for MTR, actual values of the target variables are provided to subsequent models when learning along the chain. Based on the full chain or a selected set $C = (Y_1, Y_2, \cdots, Y_d)$, the first model is constrained to establish estimates for Y_1. Subsequently, models for $Y_{i, s.t. i>1}$ are learned from a transformed data set $D_i^{'} = \left\{ \left(\mathbf{x}_i^{'(1)}, y_i^{(1)}\right), \cdots, \left(\mathbf{x}_i^{'(N)}, y_i^{(N)}\right) \right\}$, where $\mathbf{x}_i^{'(l)} = \left(x_1^{(l)}, \cdots, x_m^{(l)}, y_1^{(l)}, \cdots, y_{i-1}^{(l)}\right)$ (Borchani et al., 2015). Given such a chain of models, the output vector $\widehat{\mathbf{y}}^{(N+1)}$ for an unseen instance $\mathbf{x}^{(N+1)}$ is obtained by sequentially applying the models h_i to establish the estimates since, naturally, the actual values of the target variables are not available at prediction time.

The concept of RC was extended by establishing so-called ensemble of RC (Spyromitros-Xioufis et al., 2016) to account for the sensivity of

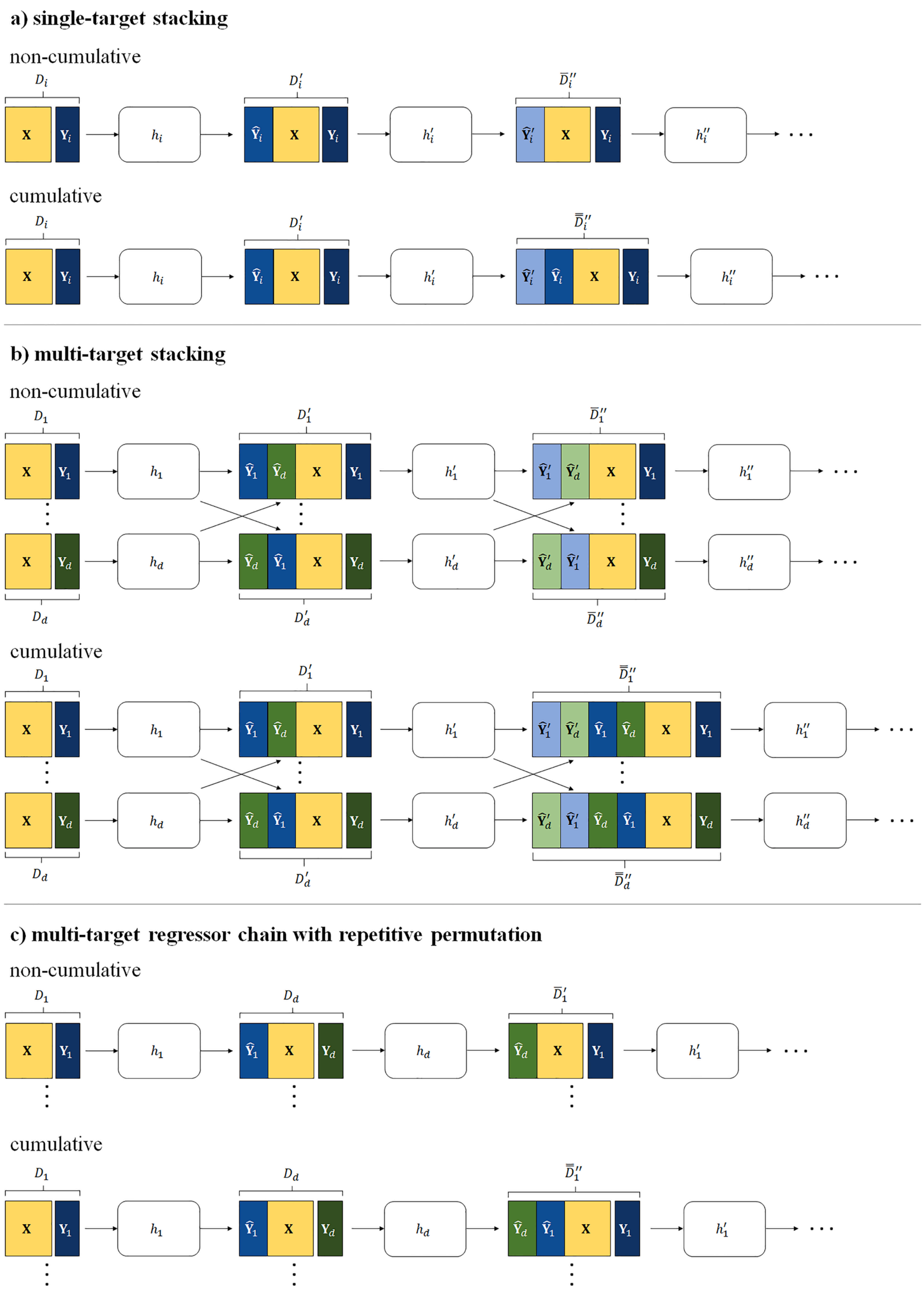

Fig. 1. Illustration of the training procedures of (a) STS, (b) MTS, and (c) ensemble of RC with repetitive permutation based on a non-cumulative and cumulative augmentation strategy of the feature vector, respectively.

the predictions with respect to the ordering of the target variables within the chain structure. Thereby, an ensemble of RC comprises multiple chains where each chain with length m consists of different and distinct permutations of the target variables. However, to keep computational complexity of building $m!$ distinct chains and learning $(m!)*m$ models low, the number of chains and elements of a chain, respectively, is constrained to 10 unless the number of target variables is equal or smaller than 3, i.e., $m! < 10$. In the latter case, all $m!$ random chains are constructed. Generally, it can be noted that out-of-sample estimates of the target variables are deployed based on an internal f-fold cross-validation procedure to avoid overfitting during training.

Existing RC methods foresee the singular occurrence of a target variable within a chain. At this point, we propose an extension of the existing methods: we equip ensembles of RC with a repetitive permutation scheme and hence allow the multiple occurrence of target variables within a chain. Thus, chains with a larger number of elements than the number of target variables can be established. To do so, first, multiple permutations with length m are drawn randomly where one target variable can occupy multiple elements of an individual chain. As such, identical objects, i.e., target variables, in the chain occur. These can be interchanged without the need to determine a new permutation. Consequently, the number of distinct permutations which can be created from m elements of the chain, whereby k are identical, is determined by $m!/k!$. Based on such a chain, estimates of the target variables are cascaded to subsequent models as additional features when learning along the chain. Consequently, a training data set is sequentially augmented by the estimates of previous target variables in the chain when learning a new model. Here, we also implement the cumulative

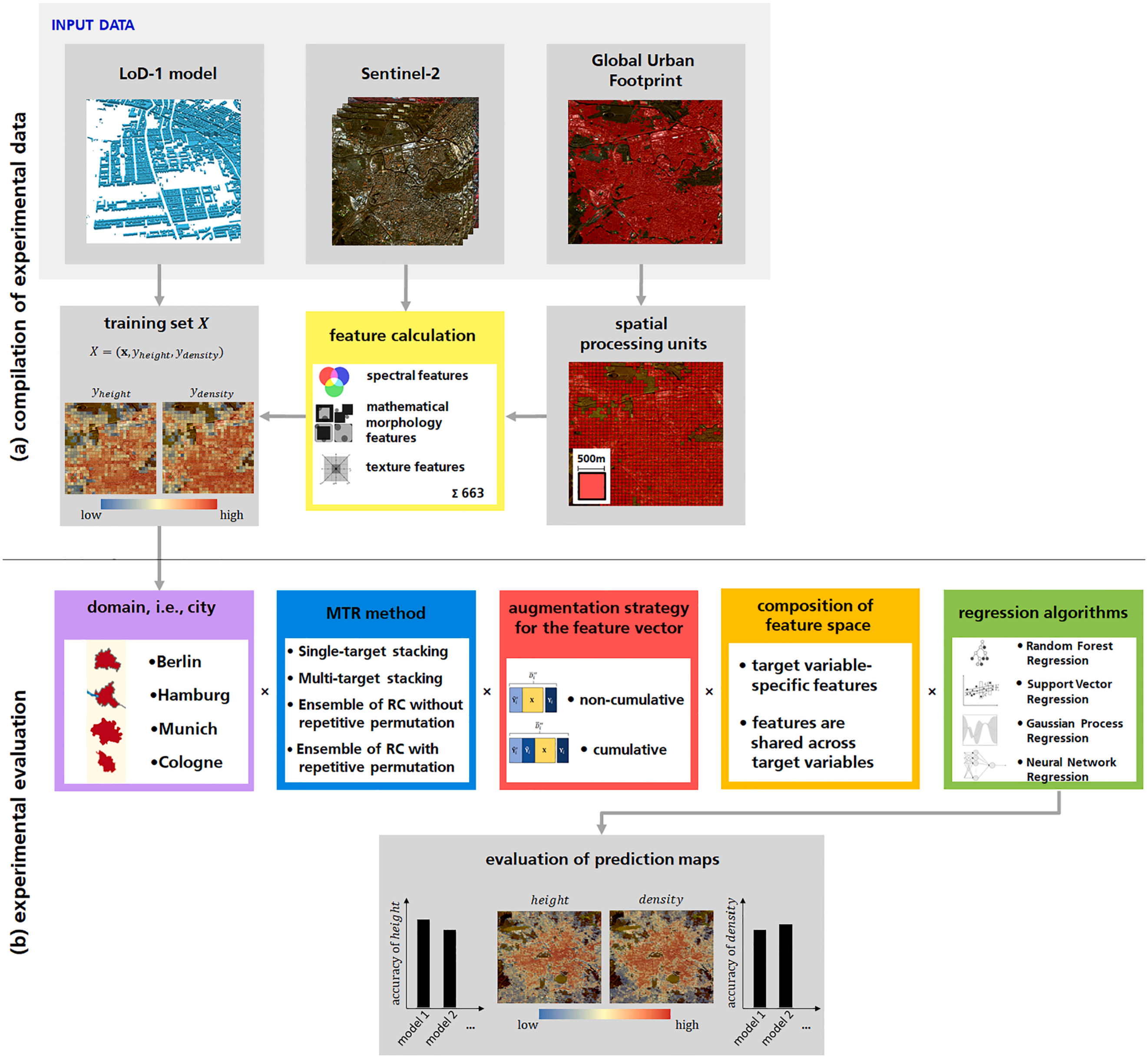
(a) compilation of experimental data
INPUT DATA
LoD-1 model
Sentinel-2
Global Urban Footprint
training set X
$X = (\mathbf{x}, y_{height}, y_{density})$
y_{height}
$y_{density}$
low
high
feature calculation
spectral features
mathematical morphology features
texture features
Σ 663
spatial processing units
500m
(b) experimental evaluation
domain, i.e., city
•Berlin
•Hamburg
•Munich
•Cologne
×
MTR method
• Single-target stacking
• Multi-target stacking
• Ensemble of RC without repetitive permutation
• Ensemble of RC with repetitive permutation
×
augmentation strategy for the feature vector
• non-cumulative
• cumulative
×
composition of feature space
• target variable-specific features
• features are shared across target variables
×
regression algorithms
• Random Forest Regression
• Support Vector Regression
• Gaussian Process Regression
• Neural Network Regression
evaluation of prediction maps
accuracy of height
model 1
model 2
...
height
density
low
high
accuracy of density
model 1
model 2
...

Fig. 2. Experimental setup. (a) The methods are deployed to estimate *built-up density* and *height* based on features from Sentinel-2 imagery. The target variables are created from LoD-1 building models and are regressed for spatial processing units which correspond to urban neighborhood scales, i.e., grid cells with an extent of 500*500 m, for the four largest German cities, i.e., Berlin, Hamburg, Munich, and Cologne. The Global Urban Footprint data set is deployed to constrain the analysis to relevant settlement areas. (b) Model accuracy is evaluated according to the considered MTR method (i.e., STS, MTS, ensemble of RC without and with repetitive permutation), composition of the feature space (i.e., deploying target variable-specific features or a set of features which is shared across all target variables), augmentation strategy for the feature vector (i.e., including previous model predictions in a non-cumulative and cumulative way), and regression algorithm (i.e., Random Forest Regression, Support Vector Regression, Gaussian Process Regression, and Neural Network Regression).

and non-cumulative case where the training data set is augmented by all and solely preceding model estimates of target variables when learning along the chain, respectively (Fig. 1c).

We followed this principle to enable a large variety of possibilities to establish a chain since it is crucial which composition of the target variables provides additional input for the prediction of Y_i. This strategy goes along with the idea of an ensemble of RC, however, by allowing repetitive permutations, we enable variable models for prediction problems with a small number of target variables. As such, the ensemble of RC with repetitive permutation scheme can be interpreted as a hybrid structure which internalizes the principles of both ensemble of RC, i.e., exploiting a chain structure with respect to the target variables, and MTS, i.e., using estimates of the target variables for new predictions.

3. Experimental setup

As can be seen from the introduction, a very relevant application field of MTR methods is the estimation of vegetation parameters with RS data. Thereby, authors exploit (potentially nonlinear) cross relations among biophysical parameters (Dey, 2021; Mandal et al., 2019; Pyo, 2019; Stojanova et al., 2010; Tuia et al., 2011). However, here, we evaluate the described methods in the context of predicting *built-up density* and *height* using features from Sentinel-2 optical satellite data. Estimating constituting variables of the urban morphology with individually learned supervised models and various geospatial data sets gained popularity recently (Geiß et al., 2020; Li et al., 2020a; Corbane, 2020; Frantz et al., 2021). Thereby, it was found that those variables can feature a certain level of correlation, also when approximated with remote sensing data (Geiß et al., 2019; Li et al., 2020a). This motivated us to exploit potential helpful dependencies of the target variables *built-up density* and *height*.

An overview on the input data and processing steps to compile the training data sets is provided by Fig. 2a. We limit the data processing to settlement areas indicated by the so-called Global Urban Footprint (GUF) layer (Esch et al., 2012). This is a binary mapping scheme which discriminates “built-up” and “non built-up” areas globally with a high spatial resolution of 12 m and classification accuracy beyond 85% (Klotz et al., 2016). Generally, the spatial resolution properties of the data hamper analyses on individual building level. The pixel spacing of ten meters of the R-G-B-NIR bands of Sentinel-2 (Drusch et al., 2012) can exceed the extent of the objects of interest (i.e., buildings). As a consequence, we work on an aggregated spatial level, i.e., we establish spatial processing units in terms of rectangular grid cells to compute *built-up density* and *height* thereof. In this study, we consider a side length of the grid cells of 500 m to reflect areas of homogeneous urban morphology (Taubenböck et al., 2016). To derive the target variables, we incorporated LoD-1 building geometries and affiliated height measurements, which are based on cadastral information (Geiß et al., 2020). Buildings are represented by extruded footprints in LoD-1 resolution (Luebke et al., 2002). *Built-up density* per grid cell is computed by establishing the ratio between the area of elevated pixels according to LoD-1 footprints and the whole settlement area as indicated by the GUF data set. *Built-up height* per grid cell is computed by extracting the median height value according to the LoD-1-based height values within a grid cell.

We implemented an exhaustive feature calculation module. Previous studies already underlined the capability of multispectral imagery to sophistically describe built-up structures (Zhang et al., 2017). The deployed optical Sentinel-2 data were subject to atmospheric corrections within the Sentinel Application Platform using the Sen2Cor module to provide level 2A products. The imagery used for feature calculation for the four cities under investigation was acquired in autumn and winter of the years 2015–2016. The dates were chosen to reduce the influence of photosynthetically active vegetation on the analysis since intra-urban vegetation frequently obscures underlying built-up structures, which then remain undetectable in the corresponding imagery. Beyond, the acquisition periods offer the possibility to exploit shadow information, which in turn encodes helpful discriminative properties of built environments. In a comparable data setup and supervised learning context, it was recently shown that building height estimations can profit more from optical than from SAR data, although a joint exploitation of both data sources can be expected to enable a further increase of prediction accuracy (Frantz et al., 2021). In order to exhaustively exploit the multispectral imagery, we compute a large number of spectral features, features related to mathematical morphology (Soille, 2004), as well as texture measures derived from the gray–level co–occurrence matrix (GLCM) (Haralick, 1979) from Sentinel-2 imagery. We deploy the same features as described in detail in (Geiß et al., 2020). Given the different feature categories and systematically varied window sizes of the deployed spatial features, overall, each spatial processing unit is represented by a 663-dimensional feature vector.

Experimental results are provided from computations regarding the settlement areas of the four largest German cities: Berlin, Hamburg, Munich, and Cologne (Fig. 2b). The processed settlement areas comprise 513 km^2 for Berlin, 577 km^2 for Hamburg, 535 km^2 for Munich, and 508 km^2 for Cologne.

The MTR methods as described in Section 2, i.e., STS, MTS, ensemble of RC with and without repetitive permutation, are evaluated. Thereby STS, MTS, and the ensemble of RC with repetitive permutation are learned over ten iterations, i.e., feature ten elements each. Regarding the latter, we draw randomly five distinct permutations. Furthermore, we evaluate two strategies regarding the composition of the feature vector. The first strategy foresees the compilation of a tailored feature vector for each target variable. We were motivated to do so since feature selection can have a valuable effect on prediction accuracy especially when dealing with feature vectors with a high dimensionality (Hughes, 1968). Consequently, we apply the filter-based correlation-based feature selection (CFS) approach (Hall, 2000; Aravena Pelizari et al., 2018) to the initial feature vector. However, in order to gain insights whether solely target variable-specific features are beneficial with respect to prediction accuracy, as an alternative, we also employ the union set of the specific features of each target variable as identified with the CFS approach and share those features across all target variables. Besides, as described in Section 2, we evaluate the results as a function of a non-cumulative and a cumulative augmentation strategy of the feature vector, respectively.

In the experiments, we considered four advanced machine learning-based regression algorithms, i.e., Random Forest Regression, Support Vector Regression, Gaussian Process Regression, and Neural Network Regression. For the Random Forest Regression models (Liaw and Wiener, 2002) we tuned the hyperparameters as follows: $n_{tree} = 500$ and $m_{try} = 1,2,\cdots,51$. For the Support Vector Regression models (Smola and Schölkopf, 2001) we deployed Gaussian RBF kernels, which take the form $K(\mathbf{x}_i\mathbf{x}_j) = \exp - \|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma2$. Learning the most appropriate SVR model in conjunction with an RBF kernel requires the definition of the regularization parameter C, the tolerance value ε, and the kernel parameter σ. Parameters were optimized according to $\sigma = \{0.01,0.02,\cdots,0.1\}$, $C = \{5,6,\cdots,15\}$, and $\varepsilon = \{0.05,0.06,\cdots,0.15\}$. Learning of the deployed Gaussian Process Regression models (Rasmussen and Williams, 2006) required tuning of hyperparameters related to the covariance or kernel functions. This includes for an RBF kernel the magnitude as well as characteristic expressions of length and noise variance. Consequently, we optimized the model according to $\sigma = \{0.01,0.02,\cdots,0.1\}$ and $\varepsilon = \{0.001,0.002,\cdots,0.1\}$. Optimization of Neural Network models required proper regularization of the weights, shape of the nonlinear function, learning rate, as well as model regularization to prevent overfitting. Moreover, the training algorithm and loss function will impact the model. In this manner, the NN models were learned based on the RSNNS hyperparameter optimization module (Bergmeier and Benitez, 2012). Model selection was carried out using the root-mean-squared error (RMSE) for all algorithms. The model which minimized RMSE within a set of iterations was selected for final prediction.

Regarding learning of regression models, training and test data were strictly spatially separated to avoid biased estimates, which can particularly occur when using spatial features due to the encoding of extrinsic spatial autocorrelation (Geiß et al., 2017). Thereby, models with randomly drawn samples, i.e., labeled grid cells, from the pool of labeled samples are learned for each city. We used 1847 training and 205 test samples for Berlin, 2079 training and 231 test samples for Hamburg, 1926 training and 214 test samples for Munich, and 1827 training and 203 test samples for Cologne, respectively. Estimated generalization capabilities of the models in the subsequent section were computed from the test samples.

4. Experimental results and discussion

To provide a first comparative overview, Fig. 3a and b show the estimations of the models over all cities and with all regressors obtained with an MTR strategy and also the estimations of the corresponding model configuration without an MTR strategy, which we refer to as initial models, for *built-up height* and *density*, respectively. The scatter plots reveal that the MTR approaches STS, MTS, and our method feature a better concentration of the density along the one-to-one line, as well as an increase in regression slope compared to the initial estimates. Thereby, the uncertainty in terms of RMSE could be reduced from 4.44 m to 3.47–3.53 m and from 7.6% to 6.11–6.42%, respectively. Typically, the models overestimate areas of low *built-up height* and *density* and underestimate areas of high *built-up height* and *density*, respectively.

More differentiated results are compiled in Fig. 4a regarding estimated *built-up height* and in Fig. 4b regarding estimated *built-up density*, respectively. Estimation accuracies are presented in terms of mean absolute error (MAE) according to the multi-target regression method, the augmentation strategy for the feature vector, the composition of the feature space, the regression algorithm, and the application domain, i.e., city.

With regard to the estimation of *built-up height*, it can be stated that generally viable accuracy levels can be achieved with the presented data setup and methodology, as the MAEs hardly exceed 3 m deviation, i.e., approximately one floor, even when solely relying on the initial model. Thereby, it can be noted that the deviation levels follow the size of a city, i.e., the deviation levels decrease from Berlin to Cologne, which follows the decreasing range of numerical height values of built-up structures of those cities.

Notably, estimates can frequently be substantially enhanced when deploying an MTR method. From a comparative model perspective, our proposed method allows lowest numerical MAE values regarding 6 out of 16 configurations and being on par 3 times with other methods. STS allows 4 times lowest MAE values while being on par 1 time, MTS allows 3 times lowest MAE values while being on par 1 time, and regressor chains without repetitive permutation are solely on par 1 time. This underlines the beneficial accuracy properties of our method. The level of achieved performance improvement is strongly related to the deployed regression algorithm. In this manner, Gaussian Process Regression and Support Vector Regression benefit most and also provide the most accurate final estimates: the best estimate for Berlin features an MAE of 1.99 m (27.3% improvement compared to the initial model) based on STS and Support Vector Regression; an MAE of 1.95 m (17% improvement) based on STS/our method and Support Vector Regression for Hamburg; an MAE of 1.36 m (32.7% improvement) based on STS and Gaussian Process Regression for Munich; and an MAE of 0.91 m (37.7% improvement) based on our method and Gaussian Process Regression for Cologne. Thereby, it also becomes traceable that the margin difference of increased accuracies between STS, MTS, and our method is frequently rather small since they all can enable a beneficial mean percentage improvement across the experiments of 14.48, 13.75, and 15.73, respectively. Nevertheless, it can be noted that other RS studies report mean percentage improvements of e.g., up to 8.63 (Stojanova et al., 2010), or 13.64 (Tuia et al., 2011), respectively, regarding their deployed algorithm adaptation-based MTR approaches and experimental setting. Hence, a substantial increase in performance can be achieved with the presented problem transformation-based methods. Solely regressors chains without repetitive permutation can hardly induce accuracy improvements in this empirical setting. This indicates that the strategy of only including a residual target variable for model learning without incorporating estimates of the actual target variable can induce model divergence in an MTR context. On the contrary, the good performance of STS suggests that the principle of utilizing preceding model outcomes of a target variable for a new prediction over multiple iterations is the main source of performance improvement here. However, the hybrid and flexible structure of ensemble of regressor chains with repetitive permutation is able to encode helpful information from both the inclusions of preceding model outcomes and a residual target variable to obtain beneficial accuracy properties of the estimates.

A likewise picture can be drawn for *built-up density* estimations. Generally, also for this target variable viable accuracy levels can be achieved since MAEs hardly indicate more than 7% deviation. The estimations can also be substantially enhanced when deploying an MTR method. From a comparative model perspective, here, our method allows lowest numerical MAE values regarding 9 out of 16 configurations and being on par 1 time with another method. In contrast, STS allows 2 times lowest MAE values while being on par 1 time, MTS allows 1 times lowest MAE values while being on par 2 times, and regressor chains without repetitive permutation allow 1 times lowest MAE values while being on par 2 times. Again, this underlines the beneficial accuracy properties of our method and also here Gaussian Process Regression and Support Vector Regression benefit most and also provide the most accurate final estimates: the best estimate for Berlin features an MAE of 3.9% (32.8% improvement compared to initial model) based on our method and Gaussian Process Regression; an MAE of 3.2% (23.8% improvement) based on our method and Support Vector Regression for Hamburg; an MAE of 4.2% (32.3% improvement) based on STS/our method and Gaussian Process Regression for Munich; and an MAE of 3.8% (32.1% improvement) based on our method and Gaussian Process Regression for Cologne. Likewise, it becomes traceable that the margin of increased accuracies between STS, MTS, and our method is frequently rather small since the corresponding mean percentage improvements across the experiments are 13.11, 12.56, and 13.33, respectively. Analogous to previous results, regressor chains without repetitive permutation do not provide substantial improvements of predictions in this experimental setting.

Regarding the augmentation strategy for the feature vector, overall, it turned out that a cumulative strategy allowed larger improvements than a non-cumulative strategy in 18 out of 32 configurations while being on par 13 times. The mean increase in accuracy of the cumulative strategy is 14.85%, while it is 12.38% for the non-cumulative strategy. Correspondingly, this favors to deploy a cumulative augmentation strategy when aiming to obtain most favorable accuracy levels. Regarding the composition of the feature space, we observed a less clear pattern. Nevertheless, a shared set of features allowed larger improvements than using solely target variable-specific features in 14 out of 32 configurations while being on par 9 times. The mean increase in accuracy with the shared set of features is 15.04%, while it is 15.01% for the target variable-specific features. Correspondingly, using a shared feature set for the target variables represented a slightly better strategy for achieving highest accuracy levels.

5. Conclusion

In this paper we have proposed the MTR method ensemble of regressor chains with repetitive permutation. We applied this method to estimate *built-up height* and *density* with features derived from Sentinel-2 data. Experimental results uncovered the beneficial performance properties of the method, since it provided most frequently the most accurate model predictions in a comparative setup. The mains findings comprise:

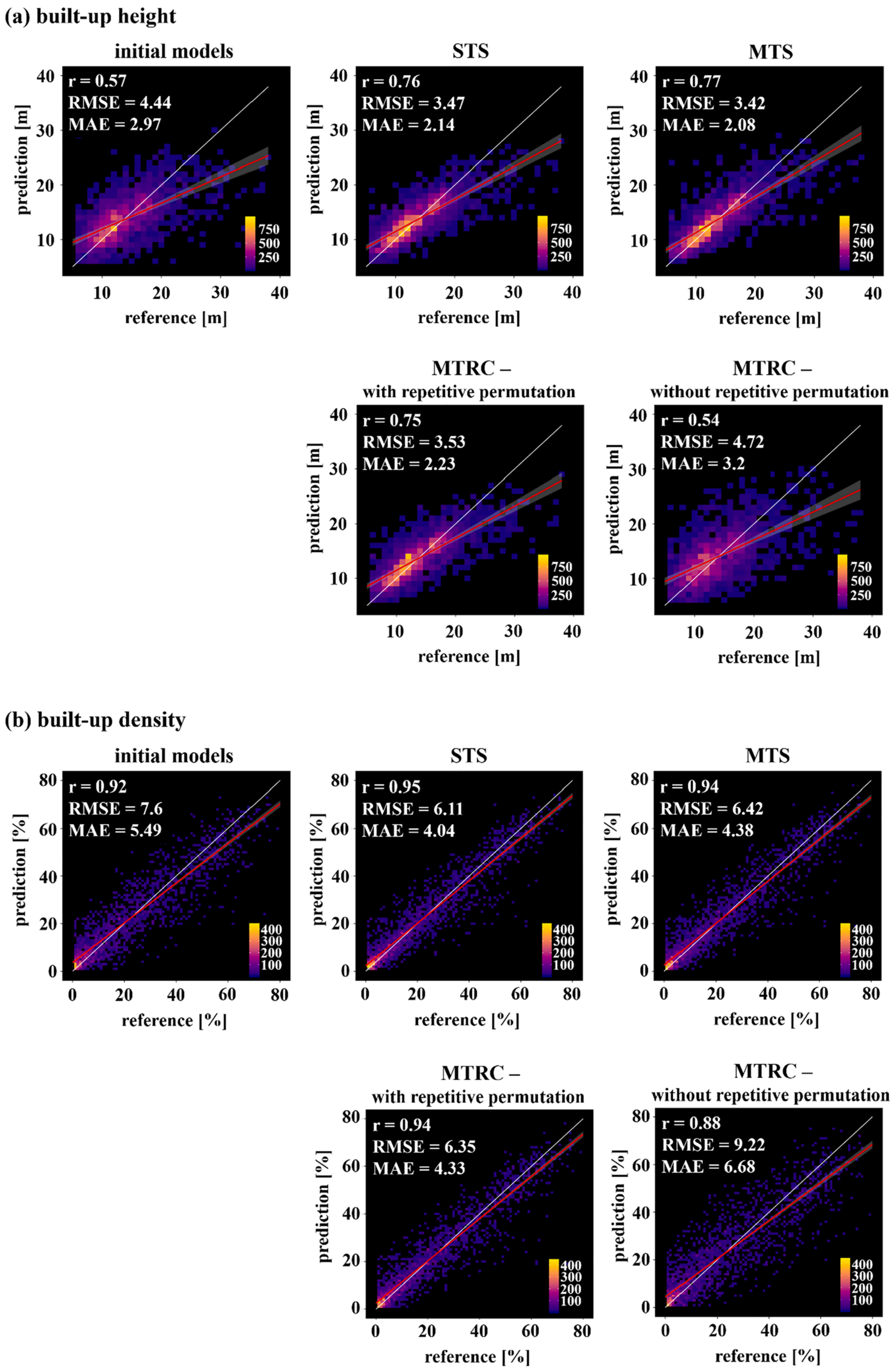

Fig. 3. Scatter plots depicting estimates of (a) *built-up height* and (b) *built-up density* obtained with the different MTR strategies including all four cities and regressors. Estimations of the corresponding model configuration without an MTR strategy, i.e., initial models, are also presented.

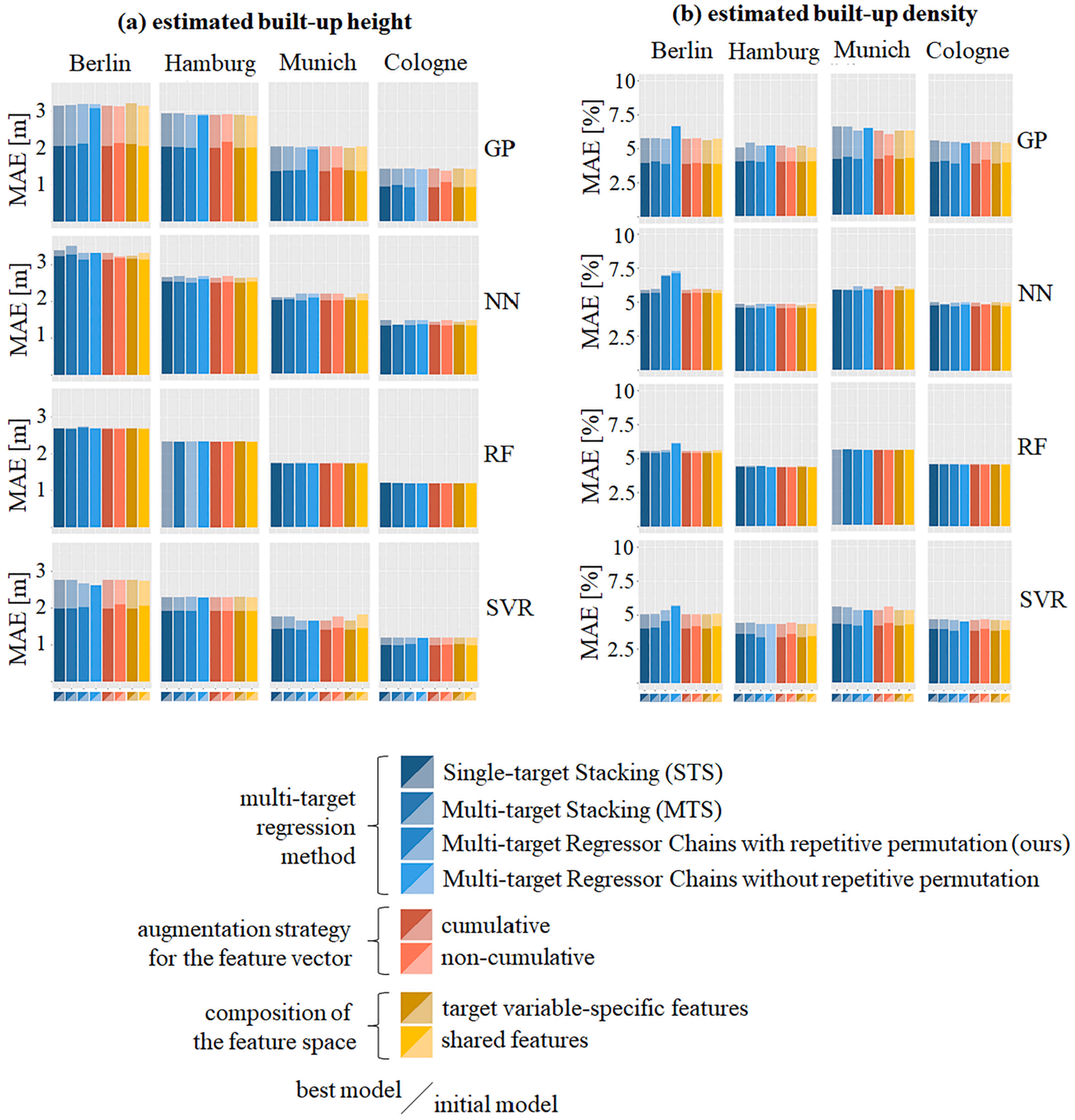

Fig. 4. MAE values regarding estimates of (a) *built-up height* and (b) *built-up density* as a function of multi-target regression method, augmentation strategy for the feature vector, composition of the feature space, regression algorithm, and application domain, i.e., city. The non-transparent bars show the estimation accuracies with an MTR method, and the semi-transparent bars show the estimation accuracies of the corresponding model configuration without an MTR strategy, i.e., initial model.

- STS, MTS, and our method turned out to be useful for substantially improving accuracy levels of initial predictions. Overall, STS, MTS, and our method could enable a mean improvement of prediction accuracy of 13,79%, 13,15%, and 14,53%, respectively.
- Gaussian Process Regression and Support Vector Regression could benefit most from the inclusion of an MTR method.
- Most accurate *built-up height* estimates feature an MAE of 2 m (27.3% improvement; STS) for Berlin, an MAE of 1.95 m (17% improvement; STS/our method) for Hamburg, an MAE of 1.36 m (32.7% improvement; STS) for Munich, and an MAE of 0.91 m (37.7% improvement; our method) for Cologne, respectively. Most accurate *built-up density* estimates feature an MAE of 3.9% (32.8% improvement; our method) for Berlin, an MAE of 3.2% (23.8% improvement; our method) for Hamburg, an MAE of 4.2% (32.3% improvement; STS/ our method) for Munich, and an MAE of 3.8% (32.1% improvement; our method) for Cologne.
- The cumulative augmentation strategy for the feature vector, which foresees the inclusion of the estimates of all preceding models, turned out to be a better strategy than including solely the estimates of the preceding model (i.e., non-cumulative strategy).
- The composition of the feature space appeared to have a less substantial effect on the accuracy levels, however, sharing a set of features across the target variables provided slightly better performance properties than using solely target variable-specific features in our experimental setup.

Overall, the beneficial accuracy properties of the considered problem transformation-based MTR methods encourage further deployment for remote sensing data analysis. Thereby, we aim to integrate ensemble of regressor chains with repetitive permutation scheme in multi-target ensemble regression models in the future.

CRediT authorship contribution statement

Christian Geiß: Conceptualization, Methodology, Software, Visualization, Writing - original draft. **Elisabeth Brzoska:** Conceptualization, Methodology, Software, Visualization, Writing - original draft. **Patrick Aravena Pelizari:** Conceptualization, Methodology, Writing - review & editing. **Sven Lautenbach:** Conceptualization, Methodology, Writing - review & editing. **Hannes Taubenböck:** Conceptualization, Methodology, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The work of Christian Geiß was supported by the Helmholtz Association under the grant “pre_DICT” (PD-305). Additionally, this research was funded in part by the German Federal Ministry of Education and Research (BMBF) under grant no. 03G0876 (project RIESGOS). We thank the two anonymous reviewers for the helpful comments on the initial version of the paper.

References

Aravena Pelizari, P., Spröhnle, K., Geiß, C., Schoepfer, E., Plank, S., Taubenböck, H., 2018. Multi-sensor feature fusion for very high spatial resolution built-up area extraction in temporary settlements. Remote Sens. Environ. 209, 793–807.

Bergmeir, C., Benitez, J.M., 2012. Neural networks in R using the Stuttgart neural network simulator: RSNNS. J. Stat. Softw. 46 (7), 1–26.

Borchani, H., Varando, G., Bielza, C., Larrañaga, P., 2015. A survey on multi-output regression. Wiley Interdisciplinary Rev.: Data Mining Knowl Discovery 5 (5), 216–233.

Breimann, L., 2001. Random forest. Mach. Learn. 45 (1), 5–32.

Camps-Valls, G., Bruzzone, L., 2005. Kernel-based methods for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 43 (6), 1351–1362.

Corbane, C., et al., 2020. Leveraging ALOS-2 PALSAR-2 for mapping built-up areas and assessing their vertical component. IEEE J. Sel. Topics Appl. Earth Obs. Remote Sens. 13, 6473–6483.

Cortes, C., Vapnik, V., 1995. Support vector networks. Mach. Learn. 20 (3), 273–297.

Dey, S., et al., 2021. BiophyNet: A Regression Network for Joint Estimation of Plant Area Index and Wet Biomass From SAR Data. IEEE Geoscience and Remote Sensing Letters 18 (10), 1701–1705.

Drusch, M., Del Bello, U., Carlier, S., Colin, O., Fernandez, V., Gascon, F., Hoersch, B., Isola, C., Laberinti, P., Martimort, P., Meygret, A., Spoto, F., Sy, O., Marchese, F., Bargellini, P., 2012. Sentinel-2: ESA’s optical high-resolution mission for GMES operational services. Remote Sens. Environ. 120, 25–36.

Esch, T., Taubenböck, H., Roth, A., Heldens, W., Felbier, A., Thiel, M., Schmidt, M., Müller, A., Dech, S., 2012. TanDEM-X mission—new perspectives for the inventory and monitoring of global settlement patterns. J. Appl. Remote Sens. 6 (1), 061702-1. https://doi.org/10.1117/1.JRS.6.061702.

Feilhauer, H., Asner, G.P., Martin, R.E., 2015. Multi-method ensemble selection of spectral bands related to leaf biochemistry. Remote Sens. Environ. 164, 57–65.

Frantz, D., Schug, F., Okujeni, A., Navacchi, C., Wagner, W., van der Linden, S., Hostert, P., 2021. National-scale mapping of building height using Sentinel-1 and Sentinel-2 time series. Remote Sens. Environ. 252, 112128. https://doi.org/10.1016/j.rse.2020.112128.

Geiß, C., Thoma, M., Pittore, M., Wieland, M., Dech, S.W., Taubenbock, H., 2017. Multitask Active Learning for Characterization of Built Environments With Multisensor Earth Observation Data. IEEE J. Sel. Topics Appl. Earth Obs. Remote Sens. 10 (12), 5583–5597.

Geiß, C., Aravena Pelizari, P., Blickensdörfer, L., Taubenböck, H., 2019. Virtual Support Vector Machines with self-learning strategy for classification of multispectral remote sensing imagery. ISPRS J. Photogramm. Remote Sens. 151, 42–58.

Geiß, C., Leichtle, T., Wurm, M., Pelizari, P.A., Standfus, I., Zhu, X.X., So, E., Siedentop, S., Esch, T., Taubenbock, H., 2019. Large-Area Characterization of Urban Morphology – Mapping of Built-Up Height and Density Using TanDEM-X and Sentinel-2 Data. IEEE J. Sel. Topics Appl. Earth Obs. Remote Sens. 12 (8), 2912–2927.

Geiß, C., Aravena Pelizari, P., Schrade, H., Brenning, A., Taubenbock, H., 2017. On the Effect of Spatially Non-disjoint Training and Test Samples on Estimated Model Generalization Capabilities in Supervised Classification with Spatial Features. IEEE Geosci. Remote Sens. Lett. 14 (11), 2008–2012.

Geiß, C., Schrade, H., Aravena Pelizari, P., Taubenböck, H., 2020. Multistrategy Ensemble Regression for Mapping of Built-up Density and Height with Sentinel-2 Data. ISPRS J. Photogramm. Remote Sens. 170, 57–71.

Geiß, C. et al., in press. Deep relearning in the geospatial domain for semantic remote sensing image segmentation. In: IEEE Geoscience and Remote Sensing Letters (in press).

Hall, M.A., 2000. Correlation-based feature selection for discrete and numeric class machine learning. In: Pat Langley (Ed.), Proceedings of the Seventeenth International Conference on Machine Learning (ICML ’00). Morgan Kaufmann Publishers Inc., San Francisco, pp. 359–366.

Haralick, R.M., 1979. Statistical and structural approaches to texture. Proc. IEEE 67 (5), 786–804.

Hughes, G., 1968. On the mean accuracy of statistical pattern recognizers. IEEE Trans. Inf. Theory 14 (1), 55–63.

Klotz, M., Kemper, T., Geiß, C., Esch, T., Taubenböck, H., 2016. How good is the map? A multi-scale cross-comparison framework for global settlement layers: Evidence from Central Europe. Remote Sens. Environ. 178, 191–212.

Kocev, D., et al., 2009. Using single- and multi-target regression trees and ensembles to model a compound index of vegetation condition. Ecol. Model. 220, 1159–1169.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.

Li, M., Koks, E., Taubenböck, H., van Vliet, J., 2020a. Continental-scale mapping and analysis of 3D building structure. Remote Sens. Environ. 245, 111859. https://doi.org/10.1016/j.rse.2020.111859.

Li, Y., Sun, H., Yan, W., Zhang, X., 2020b. Multi-output parameter-insensitive kernel twin SVR model. Neural Netw. 121, 276–293.

Liaw, A., Wiener, M., 2002. Classification and regression by randomForest. R News: Newslett. R Proj. 2 (3), 18–22.

Liu, H., Cai, J., Ong, Y.-S., 2018. Remarks on multi-output Gaussian process regression. Knowl.-Based Syst. 144, 102–121.

Luebke, D., Watson, B., Cohen, J.D., Reddy, M., Varshney, A., 2002. Level of Detail for 3D Graphics. Elsevier Science Inc.

Mandal, D., Kumar, V., McNairn, H., Bhattacharya, A., Rao, Y.S., 2019. Joint estimation of plant area index (PAI) and wet biomass in wheat and soybean from C-band polarimetric SAR data. Int. J. Appl. Earth Observ. Geoinformation 79, 24–34.

Melki, G., Cano, A., Kecman, V., Ventura, S., 2017. Multi-target support vector regression via correlation regressor chains. Inf. Sci. 415-416, 53–69.

Pal, M., 2005. Random forest classifier for remote sensing classification. Int. J. Remote Sens. 26 (1), 217–222.

Pyo, J., et al., 2019. A convolutional neural network regression for quantifying cyanobacteria using hyperspectral imagery. Remote Sens. Environ. 233, 111350.

Rasmussen, C.E., Williams, C.K.I., 2006. Gaussian Processes for Machine Learning. MIT Press, New York, NY, USA.

Read, J., Hollmen, J., 2015. Multi-label classification using labels as hidden nodes, arXiv preprint, arXiv:1503.09022.

Read, J., Pfahringer, B., Holmes, G., Frank, E., 2011. Classifier chains for multi-label classification. Mach. Learn. 85 (3), 333–359.

Reyes, O., Ventura, S., 2019. Performing multi-target regression via a parameter sharing-based deep network. Int. J. Neur. Syst. 29 (09), 1950014. https://doi.org/10.1142/S012906571950014X.

Rosentreter, J., et al., 2017. Subpixel mapping of urban areas using EnMAP data and multioutput support vector regression. IEEE JSTARS 10 (5), 1938–1948.

Sanchez-Fernandez, M., de Prado-Cumplido, M., Arenas-Garcia, J., Perez-Cruz, F., 2004. SVM multiregression for nonlinear channel estimation in multiple-input multiple-output systems. IEEE Trans. Signal Process. 52 (8), 2298–2307.

Santana, E.J., dos Santos, F.R., Mastelini, S.M., Melquiades, F.L., Barbon, S. Jr, Improved prediction of soil properties with multi-target stacked generalisation on EDXRF spectra. Preprint arXiv:2002.04312.

Segal, M., Xiao, Y., 2011. Multivariate random forests. Data Mining Knowl. Discov. 1 (1), 80–87.

Smola, A., Schölkopf, B., 2001. A tutorial on support Vector regression. Statist. Comput.

Soille, P., 2004. Morphological Image Analysis: Principles and Applications. Springer.

Souza da Silva, B., 2020. Outlier robust extreme learning for multi-target regression. Expert Syst. Appl. 140, 112877.

Spyromitros-Xioufis, E., Tsoumakas, G., Groves, W., Vlahavas, I., 2016. Multi-target regression via input space expansion: treating targets as inputs. Mach. Learn. 104 (1), 55–98.

Stojanova, D., Panov, P., Gjorgjioski, V., Kobler, A., Džeroski, S., 2010. Estimating vegetation height and canopy cover from remotely sensed data with machine learning. Ecol. Inf. 5 (4), 256–266.

Taubenböck, H., et al., 2016. The physical density of the city – deconstruction o the delusive density measure with evidence from two European megacities. ISPRS Int. J. Geo-Inf. 5 (11).

Tsoumakas, G., Katakis, I., 2007. Multi-label classification: an overview. Int. J. Data Warehous. Min. 3 (3), 1–13.

Tuia, D., Verrelst, J., Alonso, L., Perez-Cruz, F., Camps-Valls, G., 2011. Multioutput support vector regression for remote sensing biophysical parameter estimation. IEEE Geosci. Remote Sens. Lett. 8 (4), 804–808.

Wolpert, D.H., 1992. Stacked generalization. Neural Netw. 5 (2), 241–259.

Zhang, T., Huang, X., Wen, D., Li, J., 2017. Urban building density estimation from high-resolution imagery using multiple features and support vector regression. IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens. 10 (7), 3265–3280.

Further reading

Li, X., et al., 2021. RMoR-Aion: robust multioutput regression by simultaneously alleviating input and output noises. IEEE Transactions on Neural Networks and Learning Systems 1351–1364.