

# Performance evaluation and hyperparameter tuning of statistical and machine-learning models using spatial data

Patrick Schratz<sup>a</sup>, Jannes Muenchow<sup>a</sup>, Eugenia Iturritxa<sup>b</sup>, Alexander Brenning<sup>a</sup>

<sup>a</sup>*Department of Geography, GIScience group, Grietgasse 6, 07743, Jena, Germany*

<sup>b</sup>*NEIKER, Granja Modelo Arkaute, Apdo. 46, 01080 Vitoria-Gasteiz, Arab, Spain*

---

## Abstract

This template helps you to create a properly formatted L<sup>A</sup>T<sub>E</sub>X manuscript.

*Keywords:* spatial modeling, machine learning, model selection, parameter tuning, spatial cross-validation

---

## 1. Introduction

Statistical learning has become one of the most important tools in the era of knowledge **building** from big data in fields as diverse as business (finance, geomarketing) (Schernthanner et al., 2017; Heaton et al., 2016), astrophysics  
5 (Garofalo et al., 2016), medicine (Leung et al., 2016), the public sector (Maenner et al., 2016) and the sciences. We can classify statistical learning broadly into supervised (statistical models, machine learning) and unsupervised techniques (ordination, clustering) (James et al., 2013b). Though both **streams** are important in the **spatial sciences**, we will focus in this paper on spatial predic-  
10 tions using and comparing statistical models and machine learning techniques. Spatial **predictions** are of **utmost** importance in a wide variety of fields. This includes geomorphology (Brenning et al., 2015), remote sensing (Stelmaszczuk-Górska et al., 2017), hydrology (Naghibi et al., 2016), epidemiology (Adler et al.,

---

\*Corresponding author

*Email address:* [patrick.schratz@uni-jena.de](mailto:patrick.schratz@uni-jena.de) (Patrick Schratz)

2017), climatology (Voyant et al., 2017), the soil sciences (Hengl et al., 2017) and  
15 of course ecology. Ecological applications range from species distribution models  
(Muenchow et al., 2013a), predicting floristic (Muenchow et al., 2013b) and fau-  
nal composition, the influence of climate change, e.g. range shifts (REFERENZ  
Dieker) and disentangling the relationships between species and their environ-  
ment (Muenchow et al., 2013c), and disease mapping as for example caused by  
20 fungal infections (Iturrity et al., 2014) to biomass estimation (Fassnacht et al.,  
2014) and species distribution modeling (Wieland et al., 2017; Halvorsen et al.,  
2016).

Fungal species such as *Diplodia pinea* inflict severe damage to *Pinus radiata*  
trees (Wingfield et al., 2008). Infected forest stands cause economic as well as  
25 ecological damages worldwide (Ganley et al., 2009). In Spain, the local econ-  
omy highly depends on the production of timber from Monterrey Pine (*Pinus*  
*radiata*). About 25% of Spain's timber production stems from *Pinus radiata*  
plantations in northern Spain, and here mostly from the Basque Country (Itur-  
ritxa et al., 2014). Consequently, the early detection and subsequent contain-  
30 ment is vital to the survival of forest stands. Statistical and machine-learning  
models provides the means to do so.

Consequences of *Diplodia pinea* infestation include shoot blightness of seedlings  
or trees, emergence of stem cankers and more (Iturrity et al., 2014). *Fusarium*  
*circinatum* mainly attacks branch tips leaving behind desiccation symptoms.  
35 This pathogen also targets trunks and tissue which frequently leads to canker  
outbreaks (Iturrity et al., 2014). Certain climatic conditions greatly favor the  
dispersal of pathogens (Wingfield et al., 2008). For instance, temperature, pre-  
cipitation and trees previously damaged by hail were the most important vari-  
ables when predicting the probability of tree infection in the Basque country  
40 (Iturrity et al., 2014).

Generally, statistical models allow the interpretation of coefficients. This  
should be certainly the main decision criteria when it comes to analyzing the  
relationship between a response such as species richness or a single species and  
the corresponding environment (Goetz et al., 2015). Machine learning tech-

45 niques have gained **in** popularity due to their ability to handle high-dimensional and highly correlated data, the lack of underlying model assumptions and user-friendly implementations in widely used data analysis software. Some model comparison studies **within the spatial community** showed that machine learning models might be the better choice when the aim is predictive accuracy  
50 (Smoliński & Radtke, 2016; Hong et al., 2015; Youssef et al., 2015). However, others found no major performance difference to statistical models (Goetz et al., 2015; Bui et al., 2015).

Most spatial modeling studies do not use (spatial) cross-validation to assess  
model performance (Youssef et al., 2015; Wollan et al., 2008; Ward, 2006; Wang  
55 et al., 2007; Hobbelen et al., 2010; Bui et al., 2015; Hong et al., 2015). (Smoliński & Radtke, 2016) used Cross-Validation (CV) but with a non-spatial partitioning setting. Goetz et al. (2015) and Ruß & Brenning (2010) used spatial CV to compare models but did not tune hyperparameters of machine learning models. Ruß & Kruse (2010) showed the differences of spatial and non-spatial CV in the  
60 field of precision agriculture using Random Forest (RF), Support Vector Machines (SVM) and Bagging but did not tune the respective hyperparameters. Puertas et al. (2013) used both spatial cross-validation and a grid-search based hyperparameter tuning approach but did not compare multiple models. Vorpahl et al. (2012) compared statistical (Generalized Additive Model (GAM), Generalized Linear Model (GLM) Multivariate Adaptive Regression Splines (MARS)  
65 and machine-learning (Artificial Neural Network (ANN), Classification and Regression Trees (CART), RF, Boosted Regression Trees (BRT), Maximum Entropy Model (MEM)) models using only non-spatial CV and no tuning of hyperparameters. When only non-spatial cross-validation is used, the reported  
70 model performances are biased and overoptimistic due to the underlying spatial autocorrelation within the data (Brenning, 2005). If no hyperparameter tuning is conducted, it can not be guaranteed that the resulting predictive accuracy is the best result that possibly could have been achieved by the model.

The presented **contrary** results of model comparison studies focusing on predictive performance of statistical and machine-learning models **leave** an **unclear**

conclusion whether statistical or machine-learning models show better predictive performance in spatial modeling. We propose that the main reasons of these varying results are inconsistent modeling and validation setups in combination with an unawareness of the influence of spatial autocorrelation on the predictive accuracy of models. Also, no study in the field of spatial modeling combined hyperparameter tuning with **an** comparison of spatial and non-spatial CV. With this work we present an approach on how to conduct model comparison of statistical and machine-learning models when working with spatial data that builds on three major points: (i) Awareness of the influence of spatial autocorrelation in the data and account for it, (ii) usage of (nested) spatial CV to assess model performances, (iii) whenever possible, conduct of hyperparameter tuning to ensure that the respective model is able to apply its full predictive power.

We provide the complete code in the supplementary material to make this work fully reproducible. In our exemplary analysis we used a selection of six models (statistical and machine-learning) which are commonly used in the spatial modeling community: BRT, GAM, GLM, **K-Nearest** Neighbor (KNN), RF and SVM. We investigate the effects of hyperparameter tuning, show why spatial partitioning is essential when using cross-validation to assess bias-reduced model performances when working with spatial data and analyse the resulting predictive performances.

## 2. Data and study area

### 2.1. Data

This study uses the data set from Iturritxa et al. (2014) to illustrate procedures and challenges that are common to many geospatial analyses problems. It is representative for many other ecological data sets in terms of number of observations (944) and the number (11) and mixture of type of predictors (numeric and nominal). The following (environmental) variables were used as predictors: **Temperature, precipitation, solar radiation, elevation, slope, hail, tree age, pH**

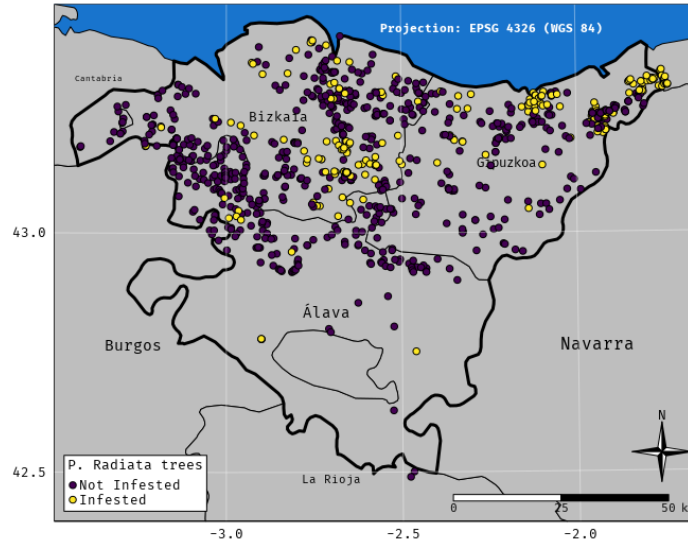


Figure 1: Spatial distribution of tree observations within the Basque Country, northern Spain, showing infection state by *Diplodia Pineae*

105 value of soil, soil type, lithology type, and the year when the tree was surveyed. Tree infection caused by fungal pathogens (here *Diplodia Pineae*) represents the response variable. We observed 224 infected and 720 healthy trees. Compared to the original data set from Iturritxa et al. (2014), we added soil types (12 classes) (Hengl et al., 2017), lithology type (17 classes) (GeoEuskadi, 1999) and

110 pH value of the soil (European Commission, 2010) to the already available predictors. Predictor 'hail', original an in-situ observation of hail damage in the work of Iturritxa et al. (2014), was substituted by a new hail variable representing the spatial distribution of hail damage potential. It was spatially modeled using a GAM with predictors being the variables of the Iturritxa et al. (2014)

115 data set. The already existing in-situ hail variable served as the reference in the validation process. The advantage of this new hail variable is its spatial availability across the Basque Country. This makes it possible to also use it for prediction purposes. This was not the case in the Iturritxa et al. (2014) study for which hail was only available as point information. We removed three obser-

120 vations due to missing information in some of the additionally added variables leaving a total of 944 observations (Table B.2). The methodology we present in this work can be easily extended to response variables with more than two classes.



## 2.2. Study area

125 The Basque country in northern Spain represents our study area (Figure 1). It has a spatial extent of 7355 km<sup>2</sup>. Precipitation decreases towards the south while the duration of summer drought increases. Mean annual precipitation ranges from 600 mm to 2000 mm with a yearly mean temperature minimum of 8°C and a maximum of 16°C (Ganuza & Almendros, 2003).

## 130 3. Methods

In this study we provide an exemplary analysis combining both: Tuning of hyperparameters using nested CV and the use of spatial CV to assess bias-reduced model performances. We compared predictive performances using three setups: (i) non-spatial CV setting with hyperparameter tuning, (ii) spatial CV  
135 setting with hyperparameter tuning, (iii) spatial CV setting without hyperparameter tuning. We used a selection of commonly used machine learning models in spatial statistical classification analyses namely RF, SVM, KNN, BRT (also known as Gradient Boosting Machine (GBM)) and statistical learning methods like GLM and GAM.

### 140 3.1. Tuning of hyperparameters

When comparing performances of models, it is important for a fair comparison to ensure that optimal (hyperparameter) settings for each model were used. While statistical modeling algorithms cannot be tuned (some perform an internal optimization, e.g. *mgcv* package), hyperparameters of machine learning  
145 algorithms need to be tuned to achieve optimal performances (Bergstra & Bengio, 2012; Hutter et al., 2011; Duarte & Wainer, 2017). In Bayesian statistics, a hyperparameter is a parameter needed to calculate a (prior) distribution of



another parameter (Bernardo & Smith, 2009). In the context of modeling the term 'parameter' is used if such are directly fitted to the data (e.g. regression coefficients) whereas 'hyperparameters' are determined by some penalization procedure like CV. In practice we often see the following: (i) Inexperienced users often start by manually trying different hyperparameter values and checking the performance of the resulting model. This time consuming approach will most likely never find the optimal parameter set, especially if the tuning ranges of specific hyperparameters are large (e.g. for SVM). It is referred to as 'manual search' (Bergstra & Bengio, 2012). (ii) A more commonly used approach is to tune models using a 'grid search' (Bergstra & Bengio, 2012). A 'grid' in this context is a set of user-defined characteristics of hyperparameter combinations. All possible combinations of the hyperparameters to be tuned will be checked on the model. This approach has some limitations: Expert knowledge about meaningful combinations is needed and it quickly leads to computational problems if the search space needs to cover more than two hyperparameters due to the exponential growth of the grid ( $y^x$ ) due to the curse of dimensionality (Bellman, 1961). As an example, three hyperparameters with each five characteristics only to test ( $3^5$ ) would already create a grid of 243 combinations. Due to its inflexibility a grid search is always dominated by other optimization procedures, e.g. by random search (Bergstra & Bengio, 2012). (iii) A random search is able to cover a large hyperparameter tuning space at relatively low cost sufficiently well (Bergstra & Bengio, 2012). Here, first a distinct number of iterations (e.g., 100) is defined. Then, for each iteration, a hyperparameter combination is randomly composed out of a user defined tuning space. A random search covers the possible tuning space uniformly. The detail of coverage is a function of feature space size and iterations chosen by the user.

We used a random search with a varying number of iterations (0, 10, 50, 100, 200, 1000) for all machine learning models in this study to detect possible tuning saturation points. The limits of the tuning spaces were set by iteratively checking the tuning results and adjusting the search space to make sure that the resulting optimal hyperparameter combinations of each fold are not possi-



bly limited by the defined search space. However, in practice this is sometimes  
 180 impossible (see the problems we faced for KNN and BRT in subsection 3.4)  
 because models start to fail if certain hyperparameter combinations are com-  
 bined. The reasons for this are often unclear but can most often be explained  
 by numerical instability. While non-convergence is no problem in general, and  
 is usually handled by the chosen modeling **framework**, these failing iterations  
 185 might consume the available computer memory while trying to converge which  
 might subsequently cause the abortion of all processes.

Cross-validation or bootstrap approaches are quite commonly used for model  
 performance evaluation and hyperparameter tuning because they provide bias-  
 reduced performance estimates (Duarte & Wainer, 2017). However, most pack-  
 190 ages currently available in R offer only random partitioning methods, assuming  
 independence of the observations. The *sperrorest* package offers functions for  
 spatial partitioning (Figure 2) but is not capable of hyperparameter tuning  
 (Brenning, 2012). Package *mlr*, which was used as the modeling framework  
 in this work, was missing spatial partitioning functions but provides a unified  
 195 framework for modeling and simplifies hyperparameter tuning. To solve this  
 conflict, we implemented the spatial partitioning methods of *sperrorest* into  
*mlr* within the work of this study.

### 3.2. Nested Cross-Validation

The **concept** of CV is to split an existing data set into training and test sets  
 200 using a user-defined number of partitions (Figure 2). The training set consists  
 of  $k - 1$  partitions with  $k$  being the number of created partitions. The model  
 is trained on the training **set** partition and evaluated on the test set partition.  
 Every observation is exactly **one time** assigned to the test set within a repetition.  
 A repetition consists of  $k$  iterations (also called 'folds') for which every time a  
 205 model is trained on the training set and evaluated on the test set.

We used nested (non-)spatial CV in in this study to assess model perfor-  
 mances. In the outer loop, a five **fold** partitioning strategy was chosen which  
 was repeated 100 times (Figure 2). For the hyperparameter tuning in the in-



ner loop, again five folds were used to split the training set of each fold into  
 210 **sub-folds**. A random search with varying number of iterations (0, 10, 50, 100,  
 200, 1000) of hyperparameter combinations was applied to each fold of the  
 inner loop. The Area Under the Receiver Operating Characteristics (ROC)  
 Curve (AUROC) was selected as goodness of fit measure due to the binomial  
 response variable. This measure combines both True Positive Rate (TPR) and  
 215 False Positive Rate (FPR) of the classification and is also independent of a specific  
 decision threshold (Candy & Breidfeller, 2013). A resulting AUROC value  
 of close to 0.5 indicates no separation power of the model (a random separation  
 would show **0.5**) while a value of 1.0 would mean that all cases were correctly  
 classified. Then, model performances were computed and averaged across folds  
 220 of the inner loop. The hyperparameter combination with the highest mean  
 AUROC tuning result across all inner loop folds was used to train a model on  
 the training set of the outer loop. This model then **got** evaluated on the test set  
 of the respective fold of the outer loop. The procedure was repeated 500 times  
 (100 repetitions with five folds each) to reduce variation introduced by parti-  
 225 tioning. See Table 1 and the respective subsections of each model for detailed  
 information on the hyperparameter tuning ranges and fitting times.

Hyperparameter tuning was performed for RF, SVM, BRT and KNN. For  
 GLM, no tuning is needed because the model has no hyperparameters and  
 assumes a linear relationship between **response** and **predictor**. For GAM, see  
 230 subsubsection 3.4.5.

### 3.3. Cross-Validation Setups

In ecology, observations are often spatially dependent. Subsequently, they  
 are affected by underlying spatial autocorrelation by a varying magnitude (Bren-  
 ning, 2005). Model performance estimates **will be** overoptimistic due to the sim-  
 235 ilarity of training and test data in a non-spatial partitioning setup when using  
 any kind of cross-validation for tuning or validation.

To showcase the difference when using spatial or non-spatial CV for model  
 performance assessment, we used the following test setups: (i) Nested non-

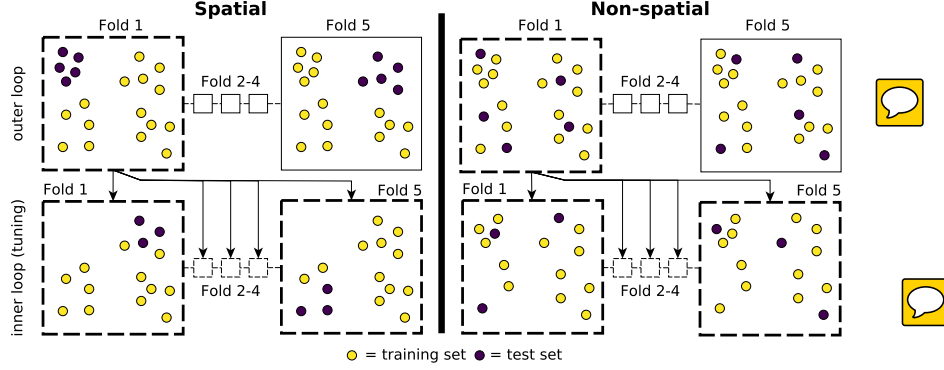


Figure 2: **Theoretical concept** of spatial and non-spatial nested cross-validation using five folds in the inner and outer loop. For every fold in the outer loop (top row), five folds with training and test data were set up using the training set of the respective outer loop as the base. The inner loop is used for tuning of hyperparameters. The winning hyperparameter combination is then trained on the respective outer loop and the performance is evaluated. The spatial setting (left) uses k-means clustering for partitioning the data and the non-spatial approach (right) a random sampling.

spatial CV which uses random partitioning (includes hyperparameter tuning),  
 240 (ii) nested spatial CV which uses k-means clustering for partitioning (Brenning,  
 2005) and results in a spatial grouping of the observations (includes hyperparam-  
 eter tuning) and (iii) spatial CV which includes no tuning of hyperparameters.  
 Setup (i) was used to show the overoptimistic results when using non-spatial CV  
 with spatial data and setup (iii) to reveal the effects of hyperparameter tuning.  
 245 Setup (ii) should be used when conducting spatial modeling.

### 3.4. Model characteristics and hyperparameters

Package selection is an often underrepresented step when conducting model-  
 ing but can have major impact on the results of the study. We attached a section  
 on package selection in Appendix A to help readers understand the **process** of  
 250 package selection in this work.

### 3.4.1. Random Forest

Classification trees are a non-linear method **which use** binary decision rules to predict a class based on the given predictors (Gordon et al., 1984). RF aggregates many classifications trees by counting the votes of all individual trees. The class with the most votes wins and will be used as the predicted class. Fitting a high number of trees is then referred to as fitting a 'forest' in a metaphorical way. Using many trees stabilizes the model (Breiman, 2001). However, RF saturates at a specific number of trees, meaning that adding more trees will not increase its performance anymore but only adds noise to the model. The **random component within** the method is that at each node in the classification tree a random number of variables (specified by parameter *mtry*) is chosen to build the tree. Also, observations are randomly selected in each tree from the data using bootstrap resampling (Breiman, 2001).

Table 1: Tuning ranges and types of model parameters. Note: We made a **pre-selection** of SVM kernels based on the best performance when using default kernel parameters (winning kernel **=** laplacedot). Subsequently, we only tuned the kernel parameters of the laplacedot kernel.

Parameter	Type	Value	Start	End	Model	Runtime(min)**
<b>C</b>	numeric	-	$2^{-12}$	$2^{25}$	SVM	590/570/0.25
<b>sigma</b>	numeric	-	$2^{-25}$	$2^6$		
kernel	<b>discrete</b>	laplacedot				
mtry	integer	-	1	11	RF	10790/5823/0.2
num.trees	integer	-	10	10000		
n.tree	integer	-	100	10000	BRT	10791/10740/0.2
shrinkage	numeric	-	0	1.5		
interaction.depth	integer	-	1	40		
k	integer	-	10	400	KNN	706/572/0.19
distance	integer	-	2	80		
kernel	<b>discrete</b>	*				

\* triangular, epanechnikov, biweight, triweight, cos, inv, gaussian, optimal

\*\* CV setting: spatial/non-spatial/spatial (no tuning); 1000 tuning iterations

### 3.4.2. Support Vector Machines

265 SVMs transform the data in a high-dimensional feature space by performing non-linear transformations of the predictor variables (Vapnik, 1998). In this high-dimensional setting, classes are separated using decision hyperplanes. Tuning of SVMs is important and not trivial due to the sensitivity of the hyperparameters across a wide search space (Duan et al., 2003).

270 We did a **pre-selection** of the kernel parameter so that we only needed to optimize hyperparameters of a single SVM model. Since all kernels come with their own kernel parameters, each SVM kernel would have had to be treated as an unique model in terms of tuning and evaluation similar to RF, KNN and all others used in this study. We selected one kernel with the aim to focus on  
275 an **inter-model** comparison instead of a SVM kernel comparison. The selection criteria for the kernel was the estimated performance (AUROC) using spatial CV with default kernel parameters. All available kernels of the *kernelab* package (rbfdot, polydot, vanilladot, tanhdot, laplacedot, besseldot, anovadot, spline-dot, stringdot) were used within this pre-selection process. Kernel laplacedot  
280 showed the best performance and was subsequently selected as the kernel for hyperparameter tuning. Performance estimates of the kernel pre-selection analysis are shown in Table B.3.



### 3.4.3. Boosted Regression Trees

BRT are different from RF in that trees are fitted on top of previous trees,  
285 i.e. in a vertical way, and not horizontally, i.e., parallel to each other. In this iterative process, each tree learns from the previous fitted trees by a magnitude specified by the *shrinkage* parameter (Elith et al., 2008). This process is also called 'stage-wise fitting' (not step-wise) because the previous fitted trees remain unchanged while additional trees are added. BRT have a tendency to-  
290 wards overfitting the more trees are added. Therefore, a combination of a small learning rate with a high number of trees is preferable. In contrast to RF, BRT fits regression trees rather than classification trees. BRT acts similar as a GLM as it can be applied to several response types (binomial, Poisson, Gaussian, etc.)




using a respective link function. Also, the final model can be seen as a large  
295 regression model with every tree being a single term (Elith et al., 2008).

#### 3.4.4. *K-Nearest Neighbor*

KNN identifies the **k-nearest** neighbors of an object to predict the target class based on the majority class among the neighbors. The first formulation of the algorithm goes back to Fix & Hodges (1951). Package *kknn* (Schliep  
300 & Hechenbichler, 2016) was used because it provides besides hyperparameter 'number of neighbors' ( $k$ ) also hyperparameter 'distance' (*distance*) and a choice between different kernels (up to 12, see Table 1). The *distance* parameter specifies the distance between the nearest neighbor and can then be used within a local **regression** analysis (which is similar to to a Locally Weighted Scatter  
305 Plot Smoothing (LOWESS) approach) to extend the standard KNN algorithm (Hechenbichler & Schliep, 2004). The original idea of the distance weighted KNN algorithm goes back to Dudani (1976).

Including weighting and kernel functions may increase predictive accuracy but can also lead to overfitting to the training data. Unlike to SVM, KNN  
310 kernels do not have tunable hyperparameters. Hence, kernels were treated as a **discrete** hyperparameter.

#### 3.4.5. *Generalized Linear Model and Generalized Additive Models*

GLMs extend linear models by allowing also **non-Gaussian** distributions, e.g., binomial, Poisson or negative binomial distributions, for the response variable.  
315 The **exponential** link between the response and the predictors already allows  for some degree of non-linearity. GAMs are an extension of GLMs allowing the response-predictor relationship to become fully non-linear. For more details please refer to Zuur et al. (2009); Wood (2006); James et al. (2013a).

We used the open-source **software** statistical programming language R (R  
320 Core Team, 2017) for all analyses and the packages *gbm* (Ridgeway, 2017) (BRT), *mgcv* (Wood, 2006) (GAM), *kernlab* (Karatzoglou et al., 2004) (SVM), *kknn* (Schliep & Hechenbichler, 2016) (KNN), and *ranger* (Wright & Ziegler,

2017) (RF). We used the *mlr* package (Bischl et al., 2016) for the tuning of hyperparameters and cross-validation. *mlr* provides a standardized interface for a wide variety of statistical and machine-learning models in R simplifying essential modeling tasks such as hyperparameter tuning, model performance evaluation and parallelization.

## 4. Results

### 4.1. Tuning iterations and runtime

For both **cases spatial** and non-spatial, RF *mtry* values from 1-3 were most often among the winning combination with **value 3 taking the lead** (Figure 4). ***num.trees* < 2500** showed the best performance in most tuning cases. Instead of an increase in predictive performance we even observed a **decrease** for RF when tuning hyperparameters *mtry* and *num.trees* (Figure 3a). Also, RF showed the second longest runtime among all chosen models. RF slowed down in runtime performance for higher iteration values compared to BRT (compare 200 vs. 1000 iterations in Figure 3b).

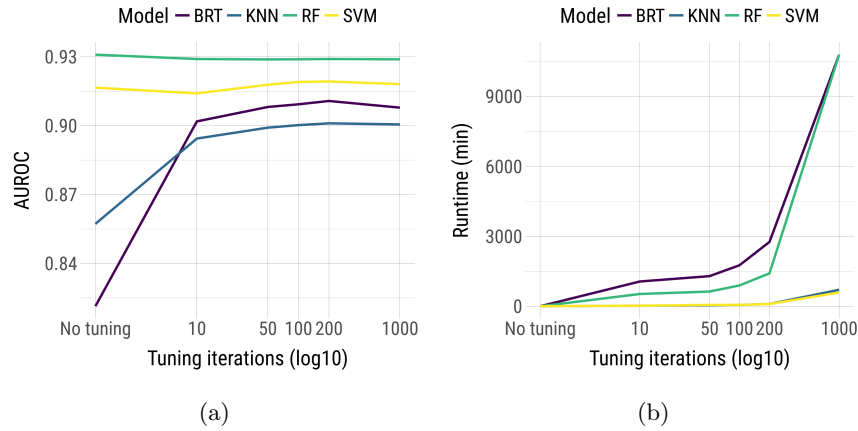


Figure 3: Hyperparameter tuning results of the spatial CV setting for BRT, KNN, RF and SVM: (a) Number of tuning iterations (1 iteration = 1 random hyperparameter combination) vs. predictive performance (AUROC) and (b) tuning iterations vs. runtime (in minutes).

SVM with 'laplacedot' kernel reveals two strong linear patterns between hyperparameters  $C$  and  $\sigma$ . If  $C$  increases,  $\sigma$  either stays at a value between  $10^{-1}$  to  $10^{-2}$  in some train/test splits or decreases linearly towards a minimum value of  $10^{-25}$ . The relationship is very stable in the non-spatial case and shows some outliers in the spatial setting (e.g.  $C \sim 10^{-8}$  and  $\sigma \sim 10^{-4.8}$ ). A small increase in predictive accuracy was found when tuning SVM. The tuning peak was found to be at 200 iterations and decreased somewhat when 1000 iterations were used (Figure 3a). SVM finished over 12 times faster than BRT and RF with 1000 tuning iterations (Figure 3b).

For BRT, a clear trend towards a small learning rate (*shrinkage*) and *num.trees* between 500 and 3000 was observed (Figure 4). This applies to both cases spatial and non-spatial. Hyperparameter *interaction.depth* was most favored between 3 - 12 if *shrinkage* was  $> 0.1$  (Figure C.6). If *shrinkage*  $< 0.1$ , the full range of *interaction.depth* between 1 - 40 was chosen roughly equally often. No pattern was observed between *num.trees* and *interaction.depth* (Figure C.6). BRT showed the highest tuning effect of all models with an increase of  $\sim 0.08$  AUROC (Figure 3a). We observed no difference between 200 and 1000 tuning iterations for BRT. BRT took the longest time to finish among all models.

For model KNN, hyperparameter  $k$  was chosen most often between 200 - 400 in combination with a low *distance* value ranging from 2-10. If  $k$  was estimated relatively low ( $< 150$ ), *distance* increased up to 80 with most values ranging around 10 - 40 (Figure 4). The most often winning kernels were "optimal" and "gaussian" (Figure C.6). A clear tuning effect is visible for KNN although the effect is as strong as for BRT (Figure 3a). KNN showed a similar runtime as SVM (Figure 3b).

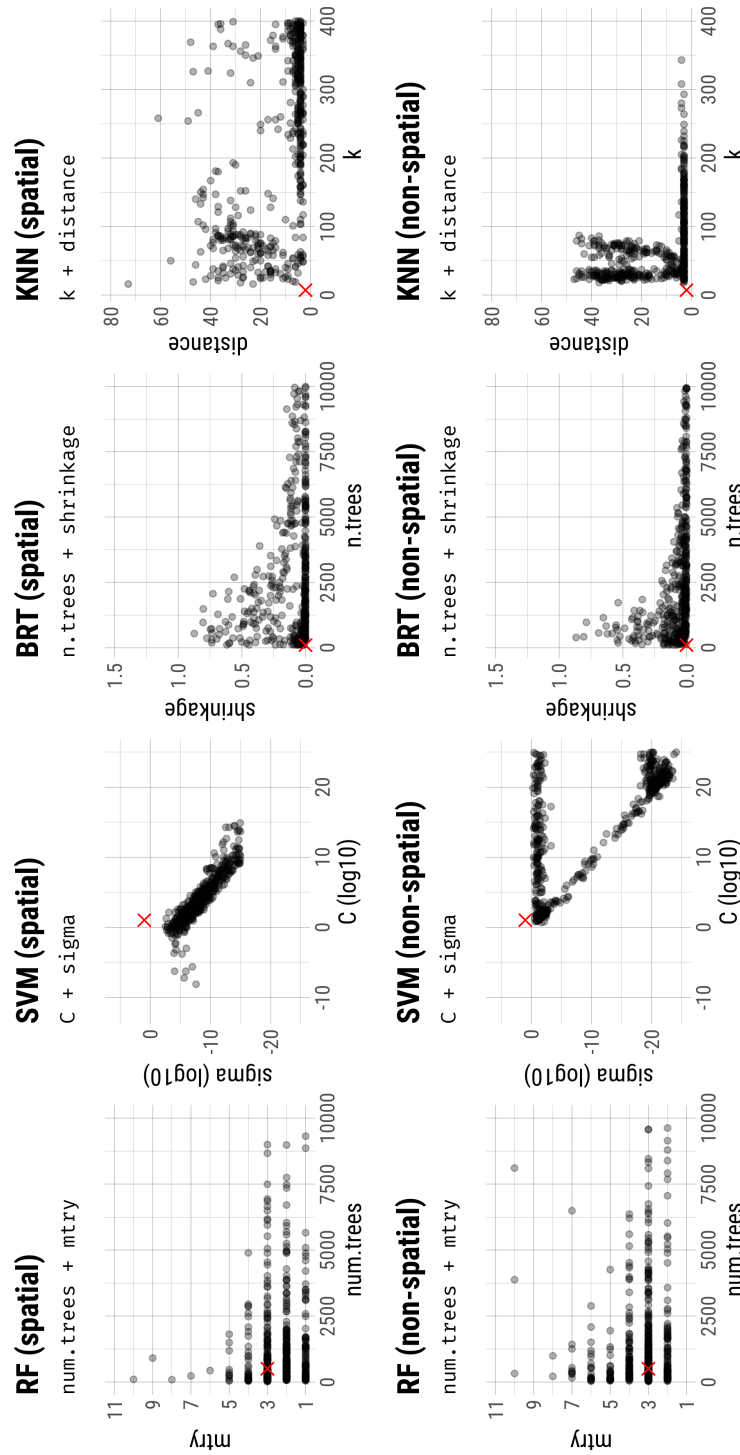


Figure 4: Best hyperparameter combinations by fold (500 **total**), each estimated from 1000 random search tuning iterations per fold using five-fold cross-validation. Split by spatial and non-spatial partitioning setup and model type. Red crosses indicate default hyperparameter values of the respective model. Black dots represent the winning hyperparameter combination out of each random search tuning of the respective fold.



#### 4.2. Predictive performance

365 BRT **win** in the non-spatial CV setup (i) but also **show** the worst performance for the spatial (no tuning) setting (iii). BRT and KNN show a high performance increase if hyperparameters are tuned (up to **~0.1** AUROC comparing setup (ii) and (iii)). In contrast, RF and SVM show almost no positive effect of hyperparameter tuning.



370 All models show overoptimistic performances for (ii) with GAM, GLM and BRT being the models deviating most (Figure 5). Statistical models (GAM, GLM) show an overall lower predictive performance between 0.05 - 0.1 AUROC compared to all machine-learning models considering the spatial (i) setting.



RF showed not only the best predictive performance but also the highest robustness due to the smallest Interquartile Range (IQR) value of all models (Figure 5). However, RF shows an up to ten times higher runtime when being tuned compared to KNN and SVM when using 1000 random search iterations (Figure 3a).

## 5. Discussion

### 380 5.1. Tuning

In theory, a **manual** search covering all possible parameter combinations would be the best approach. However, this is impractical due to required expert knowledge of the search space or computational limitations. If the **tuning dimension of hyperparameters** exceeds two, a grid search becomes impracticable (Bergstra & Bengio, 2012; Hutter et al., 2011) Since every tuning process of a model on a given data set is unique, random search provides the opportunity to tune hyperparameters without the need of expert knowledge for a suitable grid resolution as hyperparameter combinations are uniformly distributed over the search space (Bergstra & Bengio, 2012). The higher the number of iterations, the more likely it becomes that the resulting best hyperparameter combination is close to the actual optimum. Bergstra & Bengio (2012) found that random

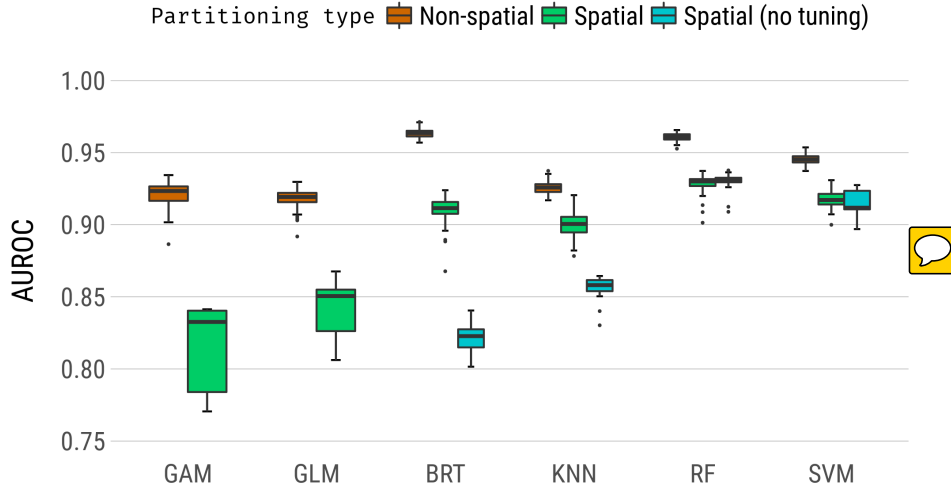


Figure 5: (Nested) spatial/non-spatial cross-validation results of BRT, GAM, GLM, KNN, RF and SVM of five folds and 100 repetitions. Boxplots represent mean AUROC values of the repetitions. For BRT, KNN, RF and SVM tuning was performed with 1000 random search iterations using a five fold partitioning setup.

search outperforms grid search in both runtime and predictive accuracy. Besides these two approaches, recently Bayesian Optimization is frequently used for optimization of black-box model (Brochu et al., 2010; Malkomes et al., 2016).

Depending on the data set characteristics, some models (e.g. RF, SVM) can be very insensitive to hyperparameter tuning (Biau & Scornet, 2016; Díaz-Uriarte & De Andres, 2006). Although we even got slightly worse predictive performances for RF, SVM for our data set when using hyperparameter tuning (Figure 3a), we recommend to always perform a tuning of hyperparameters (random search) and check the performance difference compared to the default hyperparameter settings. If no tuning is conducted, it cannot be ensured that the respective model showed its best possible predictive performance on the data set.

Computing power, especially when conducting a random search, should focus on parameter ranges that have a realistic change to cover the best parameter

combination. A practical approach that we used in this study was to iteratively check the tuning results of the nested cross-validation and adjust tuning ranges as necessary. In **theory** it should be ensured that the **main portion** of the optimum hyperparameter combination of **each outer loop fold** does not hit the  
 410 borders of the tuning space. However, this is not always possible as numerical problems within the algorithm may occur before a saturation of hyperparameters is encountered. This also happened in this study for BRT, SVM and KNN. In the case of BRT we faced the limitation of combining values greater than  $n.trees = 10000$  and  $interaction.depth = 40$ . For KNN we were unable to set  
 415 the value of  $k$  higher than 400 and for SVM the limitations were  $C = 10^{25}$  and  $sigma = 10^{-25}$ . While combinations of hyperparameters exceeding these thresholds worked in some folds, it resulted in non-recoverable errors in others which occupied all available memory of the machine while trying to converge. When extending the borders of the tuning space by the point at which numerical  
 420 instabilities are encountered, the user has to question whether this **extention** of the tuning space has a significant improvement on predictive accuracy.

The strong linear relationships between  $C$  and  $sigma$  for SVM are expected by the nature of the algorithm (Figure 4). What leaves us puzzled is the two way split in the non-spatial case. Looking at the small variance of the resulting SVM  
 425 boxplot (Figure 5) it seems to make almost no difference whether the algorithm takes a combination that follows along a gradient of  $sigma = 0$  or along a negative linear function towards  $sigma = 10^{-25}$  and  $C = 10^{25}$ . We attribute the very small increase in predictive accuracy when using hyperparameter tuning of SVM to the data set characteristics and randomness.

430 The tendency to use higher  $k$  values for KNN in the spatial tuning setting shows the influence of spatial autocorrelation (Figure 4). **If** spatial autocorrelation is stronger (in the non-spatial case), KNN mainly favored  $k < 200$  to reach a good predictive performance. In the spatial case, **way** more neighbors ( $k$ ) are needed **due to** the spatial grouping of the data (less heterogeneous data  
 435 set **characteristic**) to achieve a good predictive performance.

Tuning of hyperparameters is inevitable if the best performance of a model is

expected by the user. Depending on the model and data set characteristics the magnitude of tuning effects on the predictive performance varies. Although no significant increase or even small decreases in predictive accuracy may occur (e.g. for RF or SVM in this study), the user has to check whether a hyperparameter tuning has a positive effect on predictive performance of the model in any case.



### 5.2. Predictive Performance

In this study we compared the predictive performance of six models using three different CV setups (subsection 4.2).

The higher predictive performance of RF compared to all other models when looking at the spatial CV setup (ii) marks this model as the winner in the given model lineup. These results agree with Vorpahl et al. (2012) who also found RF being the model with the best predictive performance followed by BRT. Smoliński & Radtke (2016) also found that RF, followed by BRT and SVM, shows better predictive performance than statistical models (GAM and GLM). However, Vorpahl et al. (2012) did not use SVM within their model ensemble and both Smoliński & Radtke (2016) and Vorpahl et al. (2012) only used non-spatial CV to assess predictive accuracy. The high IQR (0.177 AUROC) of the GAM in the spatial setting (ii) is most likely related to overfitting on the training data. Predictive performance on test data becomes poor if the model overfitted on the training data which is likely due to the flexibility of the GAM (Dietterich, 1995). This is backed up by the better performance of the GLM (which assumes a simple linear relationship between response and predictors) and leads to the conclusion that the flexibility of the GAM is counter-productive for the data set used in this study and that linear approaches produce better results. Although RF clearly showed the best predictive performance in our case, SVM and BRT should always appear in a model portfolio for ecological modeling as they showed also excellent predictive power in our test case. When it comes to runtime, SVM may even be the model of choice as it outperforms RF and BRT when being tuned (Figure 3b).

We want to highlight the importance of spatial partitioning for an bias-

reduced estimate of model performance. If only non-spatial CV would have been used in this study, our main results of this study would look as follows: (i) The winning model would have been BRT instead of RF. (ii) The predictive performance would be reported with a value of  $\sim 0.96$  AUROC which is  $\sim 0.04$  AUROC higher than the bias-reduced performance estimated by spatial CV. Note that the value received using spatial CV is still overoptimistic as it is only able to reduce but not completely remove spatial autocorrelation (Brenning, 2005).

### 5.3. Other Model Evaluation Criteria

We used only one performance measure (AUROC) in this study to evaluate the predictive performance of all models. While this is also done by other model comparison studies (e.g. Goetz et al. (2015); Smoliński & Radtke (2016)), there is research on combining multiple performance measures when doing model comparison (Horn & Bischl, 2016). This approach takes multiple performances measures such as predictive measures, runtime and model sparsity into account when evaluating the suitability of a model in comparison to others.

Taking runtime and predictive performance into account, the best trade-off is achieved by SVM when performing hyperparameter tuning. However, as hyperparameter tuning does not increase predictive performance in our test case, RF would be the model of choice if no tuning is conducted. These different conclusions show the complexity in making an empirical determination about the winning model when considering multiple performance measures (here runtime and predictive performance).

Another possible model selection criteria within the spatial modeling field is the quality of the prediction surface of a resulting map. However, this point is not analysed in this study as the focus is on predictive performance. Nevertheless, it should be mentioned here because homogeneous prediction surfaces might be favored in trade-off to predictive power. Heterogeneous surfaces indicate unstable model predictions and appear when using RF for predictions (Goetz et al., 2015). GAM, GLM or SVM show much smoother prediction sur-

faces. However, such artifacts may not only rely on the algorithm itself but can be attributed to categorical variables (Goetz et al., 2015).

#### 5.4. Model Interpretability

500 If coefficients of statistical models that analyse spatial data should be interpreted, spatial autocorrelation structures should be included within the model fitting process. These ensure that model residuals are unaffected by spatial dependence. Functions like *MASS* :: *glmmPQL()* or *mgcv* :: *gamm()* provide this option. If this is ignored and coefficients of such models (e.g. GLM, GAM)  
505 are interpreted, wrong conclusions will be drawn from the results. Yet it is important to note that predictive accuracy of models without spatial autocorrelation structures is not altered. Since we only focus on predictive accuracy in this work we did not use spatial autocorrelation structures during model fitting for GLM and GAM to reduce runtime.

510 We did not focus on model interpretability in this work. However, interpretability is an important attribute of an algorithm and maybe even the most important one in ecological modeling. Ecologists often favor statistical models over machine learning models due to their ability to interpret the interactions between the predictors and the response (Goetz et al., 2011; Petschko et al.,  
515 2014). The latter are able to provide a relative estimates of variable importance but do not provide coefficients to interpret the relationships between predictors and response. In general, GLM and GAM should be favored if the main goal is to understand the dynamics in the data. Variable importance information as provided by machine learning models is only suitable to get a first idea of  
520 the data interactions but does not provide a detailed information about the predictor-response relationships. In terms of variable importance estimates of machine learning models, RF and SVM come with integrated options in their package implementations while BRT and KNN do not provide this feature. Nevertheless variable importance can also be calculated for the latter models using,  
525 for example, permutation-based variable importance approaches during cross-validation.

## 6. Conclusion

A total of six statistical and machine-learning models have been compared in this study focusing on predictive performance. For our test case, all machine learning models outperformed statistical models in terms of predictive accuracy with RF showing the highest value in combination with an insensitivity on the tuning of its hyperparameters. The effect of hyperparameter tuning of machine learning models depends on the algorithm and data set but should always be performed using a suitable amount of iterations depending on model runtime, computing infrastructure and model complexity. Spatial CV should be favored over non-spatial CV when working with spatial data to obtain bias-reduced predictive performance results. On the basis of this work we suggest to be clear on the aim before conducting spatial modeling: If the goal is to understand environmental processes, statistical models should be favored even if they do not provide the best predictive accuracy. On the other hand, if the intention is to make highly accurate spatial predictions, machine learning models should be chosen for the task. The authors hope that this work helps to fill the lack of spatial CV usage when evaluating model performances in the spatial modeling community (that apparently still exists) and serves as a starting point for spatial hyperparameter tuning of machine learning models.

## 7. Acknowledgement

## 8. Appendix

### Appendix A. Package selection

#### Appendix A.1. Random Forest

Several RF implementations exist in R. We used package *ranger* because of its fast runtime. The RF implementation in package *ranger* is up to 25 times faster (taking number of observations as benchmark criteria) and up to 60 times if hyperparameter *num.trees* is the measure to test runtime on, respectively, compared to the package *randomForest* (Wright & Ziegler, 2017).

555 Other packages such as *randomForestSRC*, *bigrf*, *RandomJungle* or *Rborist* lie in between.

#### Appendix A.2. Support Vector Machine

Package *kernlab* (Karatzoglou et al., 2004) was chosen in favor of the widely used *e1071* (Meyer et al., 2017) package because *kernlab* offers more kernel options.



#### Appendix A.3. Boosted Regression Trees

For BRT, only one implementation exists in R (to our knowledge) in package *gbm* (Ridgeway, 2017).

#### Appendix A.4. Generalized Linear/Additive Model

565 We used the base implementation of GLMs in the *stats* package which belongs to the core packages of R. For GAMs, the *mgcv* package was used in favor of *gam* because it provides several optimization methods to find the optimal smoothing degree of each variable and the ability to include random effects within the model. The *mgcv* package lets the user specify different smooth terms and limits for the degree of non-linearity (Wood, 2006). By default, the upper limit of parameter  $k$ , which limits the degree of non-linearity, is set to  $k - 1$  with  $k$  being the number of variables. Note: It is important to ensure that during optimization  $k$  does not hit the upper limit in any of the optimized smooth terms of a predictor variable. Otherwise, the degree of non-linearity of a predictor variable would be restricted and can not be modeled most accurately. Subsequently, model performance would not be optimal. Setting  $k$  to a high value relative to the final smoothing degree result leads to highly increased run-time or even convergence problems.



Appendix B. Descriptive summary of numerical and non-numerical variables


Variable	n	Min	q <sub>1</sub>	$\tilde{x}$	 $\bar{x}$	q <sub>3</sub>	Max	IQR	#NA
temp	944	12.6	14.6	15.2	15.1	15.6	16.8	1.0	0
p_sum	944	124.4	182.0	224.9	234.2	251.9	496.6	69.9	0
r_sum	944	-0.1	-0.031.0	0.009	0.0001	0.027	0.1	0.1	0
elevation	944	0.6	196.4	326.2	338.6	455.6	885.9	259.2	0
slope	944	0.3	22.1	35.4	36.6	51.3	70.0	29.2	0
age	944	1.0	9.0	15.0	16.3	21.0	40.0	12.0	0
ph	944	4.0	4.4	4.6	4.6	4.8	6.0	0.4	0

Table B.2: Descriptive summary of numerical variables.

	rbf	poly	laplace	bessel	tanh	vanilla	string	anova	spline
AUROC	0.912	0.899	0.916	0.884	0.889	0.900	nc*	nc*	nc*
* not converged									

Table B.3: Predictive performance of SVM kernels without hyperparameter tuning

Variable	Levels	n	%
diplo01	0	720	76.3
	1	224	23.7
	all	944	100.0
hail_new	0	415	44.0
	1	529	56.0
	all	944	100.0
lithology	Depsitos superficiales	32	3.4
	Areniscas	78	8.3
	Limolitas	24	2.5
	Lutitas	6	0.6
	Detrticos alternantes	428	45.3
	Margas descarbonatadas	17	1.8
	Margas	21	2.2
	Calizas impuras y calcarenitas	125	13.2
	Calizas	21	2.2
	Rocas volcnicas piroclsticas	1	0.1
	Rocas volcnicas en coladas	4	0.4
	Ofitas	5	0.5
	Arcillas con yesos y otras sales	6	0.6
	Alternancia de margocalizas margas calizas y calcarenitas	91	9.6
	Pizarras	77	8.2
	Granitos de grano grueso	4	0.4
	Granodioritas	4	0.4
	all	944	100.0
soil	Cambisols	669	70.9
	Chernozems	10	1.1
	Cryosols	6	0.6
	Durisols	6	0.6
	Ferralsols	18	1.9
	Fluvisols	7	0.7
	Gleysols	21	2.2
	Gypsisols	7	0.7
	Histosols	15	1.6
	Kastanozems	15	1.6
	Leptosols	13	1.4
	Lixisols	21	2.2
	Luvisols	136	14.4
	all	944	100.0
year	2009	402	42.6
	2010	269	28.5
	2011	109	11.6
	2012	164	17.4
	all	944	100.0

Table B.4: Non-numerical summary of predictor variables



## Appendix C. Additional hyperparameter tuning results

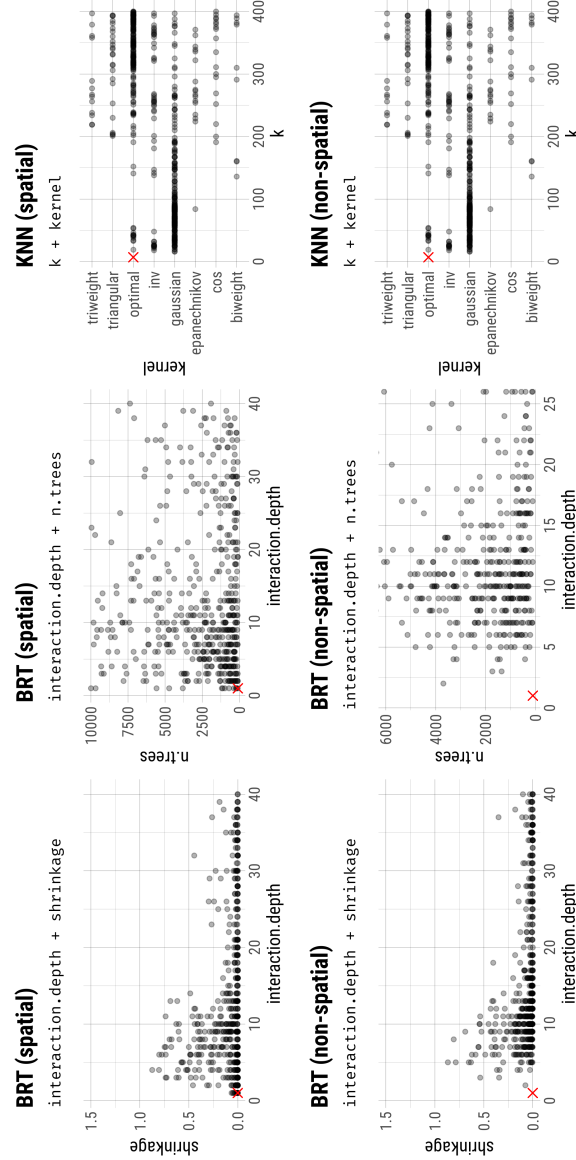


Figure C.6: Best hyperparameter combinations by fold (500 total), each estimated from 1000 random search tuning iterations per fold using five-fold cross-validation. Split by spatial and non-spatial partitioning setup and model type. Red crosses indicate default hyperparameter values of the respective model. Black dots represent the winning hyperparameter combination out of each random search tuning of the respective fold.

## References

- Adler, W., Gefeller, O., & Uter, W. (2017). Positive reactions to pairs of allergens associated with polysensitization: analysis of ivdk data with machine-learning techniques. *Contact Dermatitis*, 76, 247–251.
- Bellman, R. E. (1961). *Adaptive Control Processes*. Princeton University Press. URL: <https://doi.org/10.1515/2F9781400874668>. doi:10.1515/9781400874668.
- Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13, 281–305. URL: <http://dl.acm.org/citation.cfm?id=2188385.2188395>.
- Bernardo, J., & Smith, A. (2009). *Bayesian Theory*. Wiley Series in Probability and Statistics. Wiley. URL: <https://books.google.de/books?id=11nSgIcd7xQC>.
- Biau, G., & Scornet, E. (2016). A random forest guided tour. *TEST*, 25, 197–227. URL: <https://doi.org/10.1007/s11749-016-0481-7>. doi:10.1007/s11749-016-0481-7.
- Bischl, B., Lang, M., Kotthoff, L., Schiffner, J., Richter, J., Studerus, E., Casalicchio, G., & Jones, Z. M. (2016). mlr: Machine learning in R. *Journal of Machine Learning Research*, 17, 1–5. URL: <http://jmlr.org/papers/v17/15-066.html>.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5–32. URL: <https://doi.org/10.1023/2Fa%3A1010933404324>. doi:10.1023/a:1010933404324.
- Brenning, A. (2005). Spatial prediction models for landslide hazards: review, comparison and evaluation. *Natural Hazards and Earth System Science*, 5, 853–862. URL: <https://doi.org/10.5194/2Fnhess-5-853-2005>. doi:10.5194/nhess-5-853-2005.

- Brenning, A. (2012). Spatial cross-validation and bootstrap for the as-  
 610 sessment of prediction rules in remote sensing: The R package sperror-  
 est. In *2012 IEEE International Geoscience and Remote Sensing Sympo-  
 sium*. IEEE. URL: <https://doi.org/10.1109%2Figarss.2012.6352393>.  
 doi:10.1109/igarss.2012.6352393 R package version 2.1.0.
- Brenning, A., Schwinn, M., Ruiz-Pez, A. P., & Muenchow, J. (2015). Landslide  
 615 susceptibility near highways is increased by 1 order of magnitude in the Andes  
 of southern Ecuador, Loja province. *Natural Hazards and Earth System Sci-  
 ences*, 15, 45–57. URL: [http://www.nat-hazards-earth-syst-sci.net/  
 15/45/2015/](http://www.nat-hazards-earth-syst-sci.net/15/45/2015/).
- Brochu, E., Cora, V. M., & de Freitas, N. (2010). A tutorial on bayesian op-  
 620 timization of expensive cost functions, with application to active user mod-  
 eling and hierarchical reinforcement learning. *CoRR*, abs/1012.2599. URL:  
<http://arxiv.org/abs/1012.2599>.
- Bui, D. T., Tuan, T. A., Klempe, H., Pradhan, B., & Revhaug, I. (2015).  
 Spatial prediction models for shallow landslide hazards: a comparative as-  
 625 sessment of the efficacy of support vector machines, artificial neural net-  
 works, kernel logistic regression, and logistic model tree. *Landslides*, 13,  
 361–378. URL: <https://doi.org/10.1007%2Fs10346-015-0557-6>. doi:10.  
 1007/s10346-015-0557-6.
- Candy, J. V., & Breitfeller, E. F. (2013). *Receiver Operating Characteristic  
 630 (ROC) Curves: An Analysis Tool for Detection Performance*. Technical Re-  
 port. URL: <https://doi.org/10.2172%2F1093414>. doi:10.2172/1093414.
- Díaz-Uriarte, R., & De Andres, S. A. (2006). Gene selection and classification  
 of microarray data using random forest. *BMC bioinformatics*, 7, 3.
- Dietterich, T. (1995). Overfitting and undercomputing in machine learning.  
 635 *ACM Computing Surveys*, 27, 326–327. URL: [https://doi.org/10.1145%  
 2F212094.212114](https://doi.org/10.1145%2F212094.212114). doi:10.1145/212094.212114.

- Duan, K., Keerthi, S., & Poo, A. N. (2003). Evaluation of simple performance measures for tuning SVM hyperparameters. *Neurocomputing*, 51, 41–59. URL: <https://doi.org/10.1016%2Fs0925-2312%2802%2900601-x>.  
640 doi:10.1016/s0925-2312(02)00601-x.
- Duarte, E., & Wainer, J. (2017). Empirical comparison of cross-validation and internal metrics for tuning SVM hyperparameters. *Pattern Recognition Letters*, 88, 6–11. URL: <https://doi.org/10.1016%2Fj.patrec.2017.01.007>.  
doi:10.1016/j.patrec.2017.01.007.
- 645 Dudani, S. A. (1976). The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6, 325–327. URL: <https://doi.org/10.1109%2Ftsmc.1976.5408784>. doi:10.1109/tsmc.1976.5408784.
- Elith, J., Leathwick, J. R., & Hastie, T. (2008). A working guide  
650 to boosted regression trees. *Journal of Animal Ecology*, 77, 802–813. URL: <http://dx.doi.org/10.1111/j.1365-2656.2008.01390.x>. doi:10.1111/j.1365-2656.2008.01390.x.
- European Commission, J. R. C. (2010). 'Map of Soil pH in Europe', *Land Resources Management Unit, Institute for Environment & Sustainability*. URL:  
655 <http://esdac.jrc.ec.europa.eu/content/soil-ph-europe>.
- Fassnacht, F., Hartig, F., Latifi, H., Berger, C., Hernández, J., Corvalán, P., & Koch, B. (2014). Importance of sample size, data type and prediction method for remote sensing-based estimations of aboveground forest biomass. *Remote Sensing of Environment*, 154, 102–114. URL: <https://doi.org/10.1016%2Fj.rse.2014.07.028>.  
660 2Fj.rse.2014.07.028. doi:10.1016/j.rse.2014.07.028.
- Fix, & Hodges (1951). *Discriminatory analysis, nonparametric discrimination: Consistency properties*. Technical Report U.S. Air Force, School of Aviation Medicine, Randolph Field, TX.

- Ganley, R. J., Watt, M. S., Manning, L., & Iturrirxa, E. (2009). A global climatic  
 665 risk assessment of pitch canker disease. *Canadian Journal of Forest Research*,  
 39, 2246–2256. URL: <https://doi.org/10.1139%2F09-131>. doi:10.1139/  
 x09-131.
- Ganuza, A., & Almendros, G. (2003). Organic carbon storage in soils  
 of the Basque country (Spain): The effect of climate, vegetation type  
 670 and edaphic variables. *Biol. Fertil. Soils*, 37, 154–162. URL: 10.1007/  
 s00374-003-0579-4. doi:10.1007/s00374-003-0579-4.
- Garofalo, M., Botta, A., & Ventre, G. (2016). Astrophysics and big data: Chal-  
 lenges, methods, and tools. *Proceedings of the International Astronomical  
 Union*, 12, 345348. doi:10.1017/S1743921316012813.
- 675 GeoEuskadi (1999). *Litología y permeabilidad*. URL: <http://www.geo.euskadi.eus/geonetwork/srv/spa/main.home>.
- Goetz, J. N., Cabrera, R., Brenning, A., Heiss, G., & Leopold, P. (2015).  
 Modelling landslide susceptibility for a large geographical area using weights  
 of evidence in lower austria, austria. In *Engineering Geology for Society  
 and Territory - Volume 2* (pp. 927–930). Springer International Publish-  
 680 ing. URL: [https://doi.org/10.1007%2F978-3-319-09057-3\\_160](https://doi.org/10.1007%2F978-3-319-09057-3_160). doi:10.  
 1007/978-3-319-09057-3\_160.
- Goetz, J. N., Guthrie, R. H., & Brenning, A. (2011). Integrating physical  
 and empirical landslide susceptibility models using generalized additive mod-  
 685 els. *Geomorphology*, 129, 376–386. URL: <https://doi.org/10.1016%2Fj.geomorph.2011.03.001>.  
 doi:10.1016/j.geomorph.2011.03.001.
- Gordon, A. D., Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J.  
 (1984). Classification and regression trees. *Biometrics*, 40, 874. URL: <https://doi.org/10.2307%2F2530946>. doi:10.2307/2530946.
- 690 Halvorsen, R., Mazzoni, S., Dirksen, J. W., Næsset, E., Gobakken, T., & Ohlson,  
 M. (2016). How important are choice of model selection method and spa-



- tial autocorrelation of presence data for distribution modelling by MaxEnt? *Ecological Modelling*, 328, 108–118. URL: <https://doi.org/10.1016/j.ecolmodel.2016.02.021>. doi:10.1016/j.ecolmodel.2016.02.021.
- 695 Heaton, J. B., Polson, N. G., & Witte, J. H. (2016). Deep learning in finance. *CoRR*, abs/1602.06561. URL: <http://arxiv.org/abs/1602.06561>.
- Hechenbichler, K., & Schliep, K. (2004). Weighted k-nearest-neighbor techniques and ordinal classification. *Discussion Paper 399, SFB 386*, . URL: [https://epub.ub.uni-muenchen.de/1769/1/paper\\_399.pdf](https://epub.ub.uni-muenchen.de/1769/1/paper_399.pdf).
- 700 Hengl, T., de Jesus, J. M., Heuvelink, G. B. M., Gonzalez, M. R., Kilibarda, M., Blagotić, A., Shangguan, W., Wright, M. N., Geng, X., Bauer-Marschallinger, B., Guevara, M. A., Vargas, R., MacMillan, R. A., Batjes, N. H., Leenaars, J. G. B., Ribeiro, E., Wheeler, I., Mantel, S., & Kempen, B. (2017). Soil-Grids250m: Global gridded soil information based on machine learning. *PLOS ONE*, 12, e0169748. URL: <https://doi.org/10.1371/journal.pone.0169748>. doi:10.1371/journal.pone.0169748.
- 705 Hobbelen, P. H. F., Paveley, N. D., Fraaije, B. A., Lucas, J. A., & van den Bosch, F. (2010). Derivation and testing of a model to predict selection for fungicide resistance. *Plant Pathology*, 60, 304–313. URL: <https://doi.org/10.1111/j.1365-3059.2010.02380.x>. doi:10.1111/j.1365-3059.2010.02380.x.
- 710 Hong, H., Pradhan, B., Jebur, M. N., Bui, D. T., Xu, C., & Akgun, A. (2015). Spatial prediction of landslide hazard at the luxi area (china) using support vector machines. *Environmental Earth Sciences*, 75. URL: <https://doi.org/10.1007/s12665-015-4866-9>. doi:10.1007/s12665-015-4866-9.
- 715 Horn, D., & Bischl, B. (2016). Multi-objective parameter configuration of machine learning algorithms using model-based optimization. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE. URL: <https://doi.org/10.1109/ssci.2016.7850221>. doi:10.1109/ssci.2016.7850221.

- 720 Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. In *Lecture Notes in Computer Science* (pp. 507–523). Springer Berlin Heidelberg. URL: [https://doi.org/10.1007/978-3-642-25566-3\\_40](https://doi.org/10.1007/978-3-642-25566-3_40). doi:10.1007/978-3-642-25566-3\_40.
- Iturritxa, E., Mesanza, N., & Brenning, A. (2014). Spatial analysis of the risk  
725 of major forest diseases in Monterey pine plantations. *Plant Pathology*, 64, 880–889. URL: <http://dx.doi.org/10.1111/ppa.12328>. doi:10.1111/ppa.12328.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013a). *An Introduction to Statistical Learning*. Springer New York. URL: [https://doi.org/10.1007/](https://doi.org/10.1007/978-1-4614-7138-7)  
730 [978-1-4614-7138-7](https://doi.org/10.1007/978-1-4614-7138-7). doi:10.1007/978-1-4614-7138-7.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (Eds.) (2013b). *An introduction to statistical learning: with applications in R*. Number 103 in Springer texts in statistics. New York: Springer. OCLC: ocn828488009.
- Karatzoglou, A., Smola, A., Hornik, K., & Zeileis, A. (2004). kernlab – an S4  
735 package for kernel methods in R. *Journal of Statistical Software*, 11, 1–20. URL: <http://www.jstatsoft.org/v11/i09/>. R package version 0.9-25.
- Leung, M. K. K., DeLong, A., Alipanahi, B., & Frey, B. J. (2016). Machine learning in genomic medicine: A review of computational problems and data sets. *Proceedings of the IEEE*, 104, 176–197. doi:10.1109/JPROC.2015.2494198.
- 740 Maenner, M. J., Yeargin-Allsopp, M., Van Naarden Braun, K., Christensen, D. L., & Schieve, L. A. (2016). Development of a machine learning algorithm for the surveillance of autism spectrum disorder. *PLOS ONE*, 11, 1–11. URL: <https://doi.org/10.1371/journal.pone.0168224>. doi:10.1371/journal.pone.0168224.
- 745 Malkomes, G., Schaff, C., & Garnett, R. (2016). Bayesian optimization for automated model selection. In D. D. Lee, M. Sugiyama,

- U. V. Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 29* (pp. 2900–2908). Curran Associates, Inc. URL: <http://papers.nips.cc/paper/6466-bayesian-optimization-for-automated-model-selection.pdf>.  
750
- Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., & Leisch, F. (2017). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. URL: <https://CRAN.R-project.org/package=e1071> R package version 1.6-8.
- 755 Muenchow, J., Bruning, A., Rodriguez, E. F., & Wehrden, H. (2013a). Predictive mapping of species richness and plant species' distributions of a Peruvian fog oasis along an altitudinal gradient. *Biotropica*, 45, 557–566. URL: <http://onlinelibrary.wiley.com/doi/10.1111/btp.12049/full>.
- Muenchow, J., Feilhauer, H., Bruning, A., Rodriguez, E. F., Bayer, F., Rodriguez, R. A., & Wehrden, H. (2013b). Coupling ordination techniques and  
760 GAM to spatially predict vegetation assemblages along a climatic gradient in an ENSO-affected region of extremely high climate variability. *Journal of vegetation science*, 24, 1154–1166. URL: <http://onlinelibrary.wiley.com/doi/10.1111/jvs.12038/full>.
- 765 Muenchow, J., Hauenstein, S., Bruning, A., Bumler, R., Rodriguez, E. F., & von Wehrden, H. (2013c). Soil texture and altitude, respectively, widely determine the floristic gradient of the most diverse fog oasis in the peruvian desert. *Journal of Tropical Ecology*, 29, 427–438. doi:10.1017/S0266467413000436.
- Naghibi, S. A., Pourghasemi, H. R., & Dixon, B. (2016). Gis-based groundwater  
770 potential mapping using boosted regression tree, classification and regression tree, and random forest machine learning models in iran. *Environmental monitoring and assessment*, 188, 44.
- Petschko, H., Brenning, A., Bell, R., Goetz, J., & Glade, T. (2014). Assessing the quality of landslide susceptibility maps case study

- 775 lower austria. *Natural Hazards and Earth System Sciences*, 14, 95–  
118. URL: <https://www.nat-hazards-earth-syst-sci.net/14/95/2014/>.  
doi:10.5194/nhess-14-95-2014.
- Puertas, O. L., Brenning, A., & Meza, F. J. (2013). Balancing misclassification  
errors of land cover classification maps using support vector machines and  
780 landsat imagery in the maipo river basin (central chile, 1975–2010). *Remote  
Sensing of Environment*, 137, 112–123. URL: [https://doi.org/10.1016%](https://doi.org/10.1016%2Fj.rse.2013.06.003)  
2Fj.rse.2013.06.003. doi:10.1016/j.rse.2013.06.003.
- R Core Team (2017). *R: A Language and Environment for Statistical Com-  
puting*. R Foundation for Statistical Computing Vienna, Austria. URL:  
785 <https://www.R-project.org/> R version 3.3.3.
- Ridgeway, G. (2017). *gbm: Generalized Boosted Regression Models*. URL:  
<https://CRAN.R-project.org/package=gbm> R package version 2.1.3.
- Ruß, G., & Brenning, A. (2010). Spatial variable importance assessment for yield  
prediction in precision agriculture. In *Lecture Notes in Computer Science* (pp.  
790 184–195). Springer Berlin Heidelberg. URL: [https://doi.org/10.1007%](https://doi.org/10.1007%2F978-3-642-13062-5_18)  
2F978-3-642-13062-5\_18. doi:10.1007/978-3-642-13062-5\_18.
- Ruß, G., & Kruse, R. (2010). Regression models for spatial data: An ex-  
ample from precision agriculture. In *Advances in Data Mining. Applica-  
tions and Theoretical Aspects* (pp. 450–463). Springer Berlin Heidelberg.  
795 URL: [https://doi.org/10.1007%2F978-3-642-14400-4\\_35](https://doi.org/10.1007%2F978-3-642-14400-4_35). doi:10.1007/  
978-3-642-14400-4\_35.
- Schernthanner, H., Asche, H., Gonschorek, J., & Scheele, L. (2017). Spa-  
tial modeling and geovisualization of rental prices for real estate portals.  
*International Journal of Agricultural and Environmental Information Sys-  
800 tems*, 8, 78–91. URL: <https://doi.org/10.4018%2Fijaeis.2017040106>.  
doi:10.4018/ijaeis.2017040106.

Schliep, K., & Hechenbichler, K. (2016). *kknns: Weighted k-Nearest Neighbors*.

URL: <https://CRAN.R-project.org/package=kknns> r package version 1.3.1.

Smoliński, S., & Radtke, K. (2016). Spatial prediction of demersal fish di-

805 versity in the baltic sea: comparison of machine learning and regression-  
based techniques. *ICES Journal of Marine Science: Journal du Con-  
seil*, (p. fsw136). URL: <https://doi.org/10.1093/icesjms/fsw136>.  
doi:10.1093/icesjms/fsw136.

Stelmaszczuk-Górska, M., Thiel, C., & Schmulilius, C. (2017). Remote sens-

810 ing for aboveground biomass estimation in boreal forests. In *Earth Ob-  
servation for Land and Emergency Monitoring* (pp. 33–55). John Wi-  
ley & Sons, Ltd. URL: <https://doi.org/10.1002/2F9781118793787.ch3>.  
doi:10.1002/9781118793787.ch3.

Vapnik, V. (1998). The support vector method of function estimation. In

815 *Nonlinear Modeling* (pp. 55–85). Springer US. URL: [https://doi.org/10.1007/2F978-1-4615-5703-6\\_3](https://doi.org/10.1007/2F978-1-4615-5703-6_3).  
doi:10.1007/978-1-4615-5703-6\_3.

Vorpahl, P., Elsenbeer, H., Mrker, M., & Schrder, B. (2012). How can sta-  
tistical models help to determine driving factors of landslides? *Ecological*

*Modelling*, 239, 27–39. URL: <https://doi.org/10.1016/2Fj.ecolmodel.2011.12.007>.  
820 2011.12.007. doi:10.1016/j.ecolmodel.2011.12.007.

Voyant, C., Notton, G., Kalogirou, S., Nivet, M.-L., Paoli, C., Motte, F., &  
Fouilloy, A. (2017). Machine learning methods for solar radiation forecasting:  
A review. *Renewable Energy*, 105, 569–582.

Wang, Y.-s., Xie, B.-y., Wan, F.-h., Xiao, Q.-m., & Dai, L.-y. (2007). The

825 potential geographic distribution of *radopholus similis* in china. *Agricul-  
tural Sciences in China*, 6, 1444–1449. URL: <https://doi.org/10.1016/2Fs1671-2927%2808%2960006-1>.  
doi:10.1016/s1671-2927(08)60006-1.

Ward, D. F. (2006). Modelling the potential geographic distribution  
of invasive ant species in new zealand. *Biological Invasions*, 9,

- 830 723–735. URL: <https://doi.org/10.1007%2Fs10530-006-9072-y>. doi:10.1007/s10530-006-9072-y.
- Wieland, R., Kerkow, A., Erh, L., Kampen, H., & Walther, D. (2017). Automated feature selection for a machine learning approach toward modeling a mosquito distribution. *Ecological Modelling*, 352, 108–112. URL: <https://doi.org/10.1016%2Fj.ecolmodel.2017.02.029>. doi:10.1016/j.ecolmodel.2017.02.029.
- 835 <https://doi.org/10.1016%2Fj.ecolmodel.2017.02.029>. doi:10.1016/j.ecolmodel.2017.02.029.
- Wingfield, M. J., Hammerbacher, A., Ganley, R. J., Steenkamp, E. T., Gordon, T. R., Wingfield, B. D., & Coutinho, T. A. (2008). Pitch canker caused by *Fusarium circinatum*— a growing threat to pine plantations and forests worldwide. *Australasian Plant Pathology*, 37, 319. URL: <https://doi.org/10.1071%2Fap08036>. doi:10.1071/ap08036.
- 840 <https://doi.org/10.1071%2Fap08036>. doi:10.1071/ap08036.
- Wollan, A. K., Bakkestuen, V., Kauserud, H., Gulden, G., & Halvorsen, R. (2008). Modelling and predicting fungal distribution patterns using herbarium data. *Journal of Biogeography*, 35, 2298–2310. URL: <https://doi.org/10.1111%2Fj.1365-2699.2008.01965.x>. doi:10.1111/j.1365-2699.2008.01965.x.
- 845 <https://doi.org/10.1111%2Fj.1365-2699.2008.01965.x>. doi:10.1111/j.1365-2699.2008.01965.x.
- Wood, S. (2006). *Generalized Additive Models: An Introduction with R*. Chapman and Hall/CRC.
- Wright, M. N., & Ziegler, A. (2017). ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, 77, 1–17. doi:10.18637/jss.v077.i01.
- 850 <https://doi.org/10.18637/jss.v077.i01>. doi:10.18637/jss.v077.i01.
- Youssef, A. M., Pourghasemi, H. R., Pourtaghi, Z. S., & Al-Katheeri, M. M. (2015). Erratum to: Landslide susceptibility mapping using random forest, boosted regression tree, classification and regression tree, and general linear models and comparison of their performance at wadi tayyah basin, asir region, saudi arabia. *Landslides*, 13, 1315–1318. URL: <https://doi.org/10.1007%2Fs10346-015-0667-1>. doi:10.1007/s10346-015-0667-1.
- 855 <https://doi.org/10.1007%2Fs10346-015-0667-1>. doi:10.1007/s10346-015-0667-1.

Zuur, A. F., Ieno, E. N., Walker, N., Saveliev, A. A., & Smith, G. M.  
(2009). *Mixed effects models and extensions in ecology with R*. Springer New  
860 York. URL: <https://doi.org/10.1007/978-0-387-87458-6>. doi:10.  
1007/978-0-387-87458-6.