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

Patrick Schratz[a], Jannes Muenchow[a], Eugenia Iturritxa[b], Jakob Richter[c], Alexander Brenning[a]

[a] *Department of Geography, GIScience group, Grietgasse 6, 07743, Jena, Germany*
[b] *NEIKER, Granja Modelo –Arkaute, Apdo. 46, 01080 Vitoria-Gasteiz, Arab, Spain*
[c] *Department of Statistics, TU Dortmund University, Germany*

Abstract

Machine-learning algorithms gained popularity in recent years in the field of ecological modeling due to their promising results in predictive performance of classification problems. While the application of such algorithms has been highly simplified in the last years due to their well-documented integration in commonly used statistical programming languages such as R, there is a lot of discussion in the field of ecological modeling about non-biased performance estimation, optimization of algorithms using hyperparameter tuning and the accounting for indirect data effects such as spatial autocorrelation. In this work we compare widely used machine-learning algorithms such as boosted regression trees (BRT), k-nearest neigbor (KNN), random forest (RF) and support vector machine (SVM) to traditional parametric algorithms such as logistic regression (GLM) and generalized additive models (GAM). In detail different nested cross-validation methods are used to evaluate model performances, effects of hyperparameter tuning are investigated and pitfals when conducting spatial modeling are discussed. Pathogen infested trees in the Basque Country in Spain serve as the response variable in this work with common environmental variables such as temperature, precipitation, soil or lithology as predictors.

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

The results show that BRT and RF (0.859 and 0.863 AUROC) outperform all other methods in predictive accuracy but also have an extensive overhead in tuning time. The effect of hyperparameter tuning saturates at around 50 iterations for our data set. The difference between bias-reduced performance estimates acquired by using spatial partitioning instead of random partitioning in k-fold cross-validation is 0.087 AUROC. It is suggested to also use spatial partitioning in cross-validation for hyperparameter tuning if spatial data is present. Hyperparameters should always be tuned because they result in more robust models compared to no hyperparamter tuning, even though they do not always cause an increase in performance. Results of this study indicate that the default hyperparameters of machine-learning models may not be optimal for spatial data sets.

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

1. Introduction

Statistical learning has become an important tool in the process of knowledge building from big data in fields as diverse as business (finance, geomarketing) (Schernthanner et al., 2017; Heaton et al., 2016), astrophysics (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 (parametric models, machine learning) and unsupervised techniques (ordination, clustering) (James et al., 2013b). Though both fields are important in the spatial modeling field, we will focus in this paper on spatial predictions using and comparing parametric models and machine learning techniques. Spatial predictions are of great importance in a wide variety of fields including geomorphology (Brenning et al., 2015), remote sensing (Stelmaszczuk-Górska et al., 2017), hydrology (Naghibi et al., 2016), epidemiology (Adler et al., 2017), climatology (Voyant et al., 2017), the soil sciences (Hengl et al., 2017) and of course ecology. Ecological applications range from species distribution models

(Quillfeldt et al., 2017; Wieland et al., 2017; Halvorsen et al., 2016), predicting floristic (Muenchow et al., 2013a) and faunal composition to disentangling the relationships between species and their environment (Muenchow et al., 2013b). Further areas of applications involve biomass estimation (Fassnacht et al., 2014) and disease mapping as for example caused by fungal infections (Iturritxa et al., 2014). The latter marks the research area of this work.

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 ecological damages worldwide (Ganley et al., 2009). In Spain, the local economy 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 (Iturritxa et al., 2014). Consequently, the early detection and subsequent containment is vital to the survival of forest stands. Statistical and machine-learning models provide the means to do so.

Parametric models allow the interpretation of coefficients. This enables ecologists to interpret interactions between the response and its predictors and improve the understanding of the modeled relationship. The ability of a model to have interpretable coefficients should certainly be the main decision criteria when it comes to analyzing the relationship between a response variable such as species richness or species presence/absence and the corresponding environment (Goetz et al., 2015). Machine learning techniques have gained 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 in the spatial modeling field showed that machine learning models might be the better choice when the aim is predictive accuracy (Smoliński & Radtke, 2016; Hong et al., 2015; Youssef et al., 2015). However, others found no major performance difference to parametric models (Goetz et al., 2015; Bui et al., 2015).

When comparing models, validation methods such as (spatial) cross-validation (CV) or bootstrapping are widely used to conduct fair comparisons (Kohavi et al., 1995; Brenning, 2005). Also, it is important to tune hyperparameters of machine learning algorithms to achieve optimal performances (Bergstra & Bengio, 2012; Hutter et al., 2011; Duarte & Wainer, 2017). 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. When spatial data is present and only non-spatial CV is used, the reported model performances are biased and overoptimistic due to the underlying spatial autocorrelation within the data (Brenning, 2005). There is an increasing popularity in recent years to use spatial CV for validation when spatial data is involved (Geiß et al., 2017; Goetz et al., 2015; Ruß & Kruse, 2010; Ruß & Brenning, 2010). However, there are spatial modeling studies which do either not use (spatial) CV or similar methods to assess model performance (Youssef et al., 2015; Wollan et al., 2008; Ward, 2006; Wang et al., 2007; Hobbelen et al., 2010; Bui et al., 2015; Hong et al., 2015; Smoliński & Radtke, 2016) or leave out hyperparameter tuning Goetz et al. (2015); Ruß & Brenning (2010); Ruß & Kruse (2010); Vorpahl et al. (2012). Differing validation setups (CV/no CV/spatial CV) and tuning approaches (hyperparameter tuning/no hyperparameter tuning) make it problematic to draw general conclusions from model comparison studies. There is no current research that we know of which used a bias-reduced validation technique such as spatial CV in combination with (spatial) hyperparameter tuning to conduct a model comparison study. This work is aimed to fill this gap and should serve as an exemplary study for performing a model comparison study for spatial data that includes hyperparameter tuning and bias-reduced performance assessment. Our approach builds on two major points: (i) Awareness of the influence of spatial autocorrelation in the data and a simple approach to account for it, (ii) 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 spa-

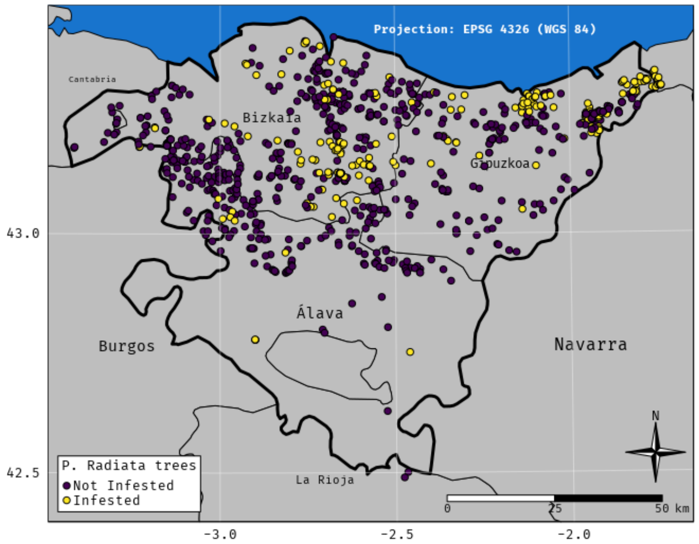


Figure 1: Spatial distribution of tree observations within the Basque Country, northern Spain, showing infection state by Diplodia Pinea

tial modeling field: Boosted Regression Trees (BRT), Generalized Additive Model (GAM), Generalized Linear Model (GLM), k-nearest neighbor (KNN), Random Forest (RF) and Support Vector Machines (SVM). We investigate the effects of hyperparameter tuning for two different partitioning settings via random search, explore the importance of spatial partitioning in cross-validation for bias-reduced model performance estimation when working with spatial data and analyze 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 observation count (944), number (11) and type numeric and nominal) of predictors. The following (environmental) variables were used as predictors: Mean temperature

(March - September), total precipitation (July - September), Potential Incoming Solar Radiation (PISR), elevation, slope, potential hail damage at trees (yes/no), tree age, pH value of soil, soil type, lithology type, and the year when the tree was surveyed. Tree infection caused by fungal pathogens (here *Diplodia Pinea*) represents the response variable. The ratio of infested and non-infested trees is roughly 3:1 (224, 720). 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 pH value of the soil (European Commission, 2010) to the already available predictors.

The predictor 'hail' represents the spatial distribution of hail damage potential at trees. Iturritxa et al. (2014) showed that hail damages serving as an entry point for pathogens is a major factor for tree infestations in the Basque Country. The hail variable of this work was spatially modeled using a GAM with predictors being the variables of the Iturritxa et al. (2014) data set. The advantage of this new hail variable is that it is spatially available across the Basque Country which makes it applicable to be used for potential prediction purposes. Before, the variable was only available as a point information.

Predictor soil is based on a regression-kriging approach with the input of 12,333 soil pH measurements from 11 different sources. The model was then predicted using 54 auxiliary variables in the form of raster maps at 1km resolution and aggregated to a spatial resolution of 5 km (European Commission, 2010).

We removed three observations due to missing information in some variables leaving a total of 944 observations (Table B.3). The methodology we present in this work can be easily extended to multiclass problems as well as to quantitative response variables.

2.2. Study area

The Basque country in northern Spain represents our study area (Figure 1). It has a spatial extent of 7355 km^2. Precipitation decreases towards the south while the duration of summer drought increases. Between 1961 and 1990, mean annual precipitation ranged from 600 to 2000 mm with annual mean temperatures between 8 and 16°C (Ganuza & Almendros, 2003).

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 four setups: Non-spatial CV with non-spatial hyperparameter tuning (*nsp/nsp*), spatial CV with spatial hyperparameter tuning (*sp/sp*), spatial CV with non-spatial hyperparameter tuning (*sp/nsp*) and spatial CV without hyperparameter tuning (*sp/not*). 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 the statistical learning methods GLM and GAM.

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 are used. While statistical modeling algorithms cannot be tuned (although some perform an internal optimization, e.g. *mgcv* package), hyperparameters of machine learning 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 optimizing CV estimates of model performance.

In practice we often see the following: (i) Inexperienced users usually start by manually trying different hyperparameter values and checking the performance of the fitted model. This time consuming approach will most likely never find the optimal parameter set, especially if the hyperparameter is a numeric one (e.g. for SVM). This approach 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 hyperparameter settings. Unless specified differently, the algorithm will be executed with all theoretically possible settings of hyperparameter to find the best setting based on a performance measure. This approach has some limitations: In practice, expert knowledge about meaningful grid settings 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) that is caused by 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 dominated by other optimization procedures, e.g. 'random search' (Bergstra & Bengio, 2012). (iii) A random search is able to cover a large hyperparameter

Table 1: Hyperparameter limits and types for each model. Notations of hyperparameters from the respective R packages were used.

Model (package)	Hyperparameter	Type	Value	Start	End
SVM (kernlab)	`C`	numeric	-	2^{-12}	2^{15}
	σ	numeric	-	2^{-15}	2^{6}
	`kernel`	nominal	`rbfdot`		
RF (ranger)	`mtry`	integer	-	1	11
	`num.trees`	integer	-	10	10000
BRT (gbm)	`n.tree`	integer	-	100	10000
	`shrinkage`	numeric	-	0	1.5
	`interaction.depth`	integer	-	1	40
KNN (kknn)	`k`	integer	-	10	400
	`distance`	integer	-	2	80
	`kernel`	nominal	*		

* triangular, Epanechnikov, biweight, triweight, cos, inv, Gaussian, optimal

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 setting is randomly composed out of a user defined tuning space. When using random search, hyperparameter settings are drawn randomly from a uniform distribution within the search space. Parameter limits and number of iterations need to be specified in order to apply this method.

We used a random search with a varying number of iterations (10, 50, 100, 200, 1000) for all machine learning models in this study to analyse the difference of varying tuning iterations. In addition, all models were fitted using their respective default hyperparameter settings, i.e. no tuning was performed. For SVM we used $\sigma = 1$ and $C = 1$ to suppress the automatic tuning of the *kernlab* package. The ranges 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 settings of each fold are not possibly limited by the defined search space. However, in practice this is sometimes impossible (see the problems we faced for KNN and BRT in subsection 3.4) because models start to fail if limits extend certain parameter limits.

CV or bootstrap approaches are quite commonly used for model performance evaluation and hyperparameter tuning because they provide bias-reduced performance estimates and allow to asses the ability of a model to generalize from (spatial) data (Duarte & Wainer, 2017; Brenning, 2005). However, most packages offering CV solutions in R offer only random partitioning methods, assuming independence of the observations. The *sperrorest* package offers functions for spatial partitioning (Figure 2) and (spatial) CV but has no integrated option to tune hyperparameters (Brenning, 2012). Package *mlr*, which was used as the modeling framework in this work, was missing spatial partitioning functions but provides a unified framework for modeling and simplifies hyperparameter tuning. Within the work of this study we implemented the spatial partitioning methods of *sperrorest* into *mlr*.

3.2. Nested Cross-Validation

The idea of CV is to split an existing data set into training and test sets using a user-defined number of partitions (Figure 2). First, the data set is divided in `k` partitions. The training set consists of $k - 1$ partitions and the test set of the remaining partition. The model is trained on the training partition and evaluated on the test partition. A repetition consists of k iterations (also called 'folds') for which every time a model is trained on the training set and evaluated on the test set. Each partition serves as a test set once.

In ecology, observations are often spatially dependent (Legendre & Fortin, 1989). Subsequently, they are affected by underlying spatial autocorrelation by a varying magnitude (Brenning, 2005). Model performance estimates will most often be overoptimistic due to the similarity of training and test data in a non-spatial partitioning setup when using any kind of cross-validation for tuning or validation (Brenning, 2012). Therefore, spatial cross-validation should be used in any kind of performance evaluation when spatial data is involved. In contrast to non-spatial CV, spatial CV reduces the influence of spatial autocorrelation,

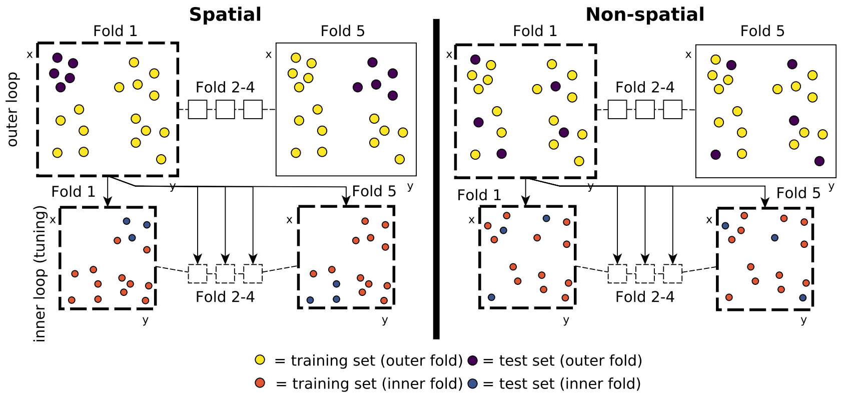

Spatial
Non-spatial
outer loop
inner loop (tuning)
Fold 1
Fold 5
Fold 2-4
x
y
● = training set (outer fold)
● = test set (outer fold)
● = training set (inner fold)
● = test set (inner fold)

Figure 2: Theoretical concept of spatial and non-spatial nested cross-validation using five folds in the inner and outer loop. Yellow/purple dots represent the training and test set in the outer loop, respectively. The inner loop is based on the respective outer loop fold and consists again of training (orange) and test set (blue).

that is present in spatial data, by partitioning the data into spatially disjoint subsets (Figure 2).

In the outer loop, a five-fold partitioning strategy was chosen which was repeated 100 times (Figure 2). For the hyperparameter tuning in the inner loop, again five folds were used to split the training set of each fold. A random search with a varying number of iterations (0, 10, 50, 100, 200, 1000) was applied to each fold of the inner loop. The Area Under the Receiver Operating Characteristics (ROC) Curve (AUROC) was selected as a goodness of fit measure due to the binary response variable. The present methodology can also be applied with other skill scores which are suited for binary classification. This measure combines both True Positive Rate (TPR) and False Positive Rate (FPR) of the classification and is also independent of a specific decision threshold (Candy & Breitfeller, 2013). A resulting AUROC value of close to 0.5 indicates no separation power of the model while a value of 1.0 would mean that all cases were correctly classified. Then, model performances were computed and averaged across folds of the inner loop. The hyperparameter setting 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 was 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 the variance introduced by partitioning. See Table 1 and the respective subsections of each model for detailed information on the hyperparameter ranges and calculation 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 logit relationship between response and predictors. For GAM, see subsubsection 3.4.5.

3.3. Cross-Validation Setups

To showcase the difference when using spatial or non-spatial CV for model performance assessment, we used the following CV setups: (*nsp/nsp*) Nested non-spatial CV which uses random partitioning (including non-spatial hyperparameter tuning), (*nsp/nsp*) nested spatial CV which uses k-means clustering for partitioning (Brenning, 2005) and results in a spatial grouping of the observations (including non-spatial hyperparameter tuning), (*sp/sp*) nested spatial CV including spatial hyperparameter tuning and (*sp/not*) spatial CV without hyperparameter tuning. Setup (*nsp/nsp*) was used to show the overoptimistic results when using non-spatial CV with spatial data and setups *nsp/nsp*, *sp/sp* to reveal the effects of hyperparameter tuning. Setup (*sp/sp*) should be used when conducting spatial modeling.

Runtime was estimated on a server running a Debian 9 operating system. All available 48 cores were used during CV and tuning. Processes ran in parallel on the tuning level.

3.4. Model characteristics and hyperparameters

Package selection is often an underrepresented step when conducting modeling but can have major impact on the results of the study. We attached a section about package selection in Appendix A to give readers the opportunity to comprehend our package selections.

3.4.1. Random Forest

'Classification trees' are a non-linear concept 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 increases computing time. Randomness is introduced by selecting a random subset of variables at each node in the classification tree to build the tree (specified by parameter m_{try}). Also, observations are randomly selected in each tree from the data using bootstrap samples (Breiman, 2001).

3.4.2. Support Vector Machines

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).

We decided to use the Radial Basis Function (RBF) kernel (also known as Gaussian kernel) which is the default in most implementations and most commonly used in the literature (Meyer et al., 2017; Guo et al., 2005; Pradhan, 2013). An exploratory analysis of the Laplace and Bessel kernels was done including respective hyperparameter tuning. All these kernels (including the RBF kernel) are classified as ”general purpose kernels” (Karatzoglou et al., 2004).

3.4.3. Boosted Regression Trees

BRT are different from RF in that trees are fitted on top of previous trees instead of being fitted parallel to each other without a relation to adjacent trees. 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 towards overfitting the more trees are added. Therefore, a combination of a small learning rate with a high number of trees is preferable. 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 regression model with every tree being a single term (Elith et al., 2008).

3.4.4. k-Nearest Neighbor

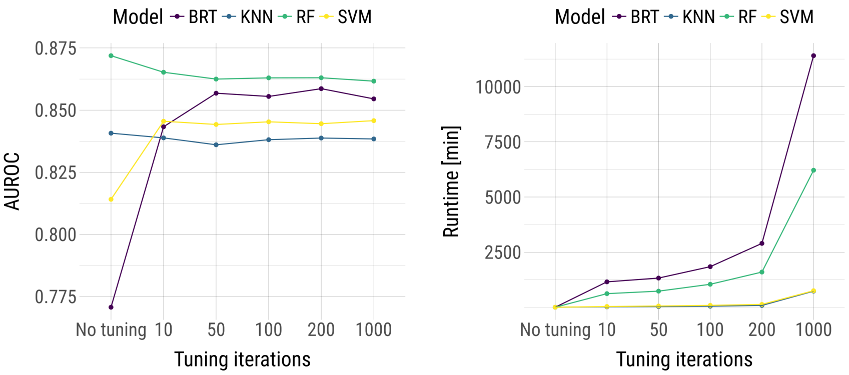
KNN identifies the K-nearest neighbors within the training set for a new observation to predict the target class based on the majority class among the neighbors. The first formulation of the algorithm goes back to Fix & Hodges (1951). Package *kknn* (Schliep & Hechenbichler, 2016) was used because it provides besides the hyperparameter *number of neighbors* ($n_{neighbors}$) also a hyperparameter that allows to set the parameter of Minkowski distance (*dist*) and a choice between different kernels (up to 12, see Table 1). Training observations that are more close to the prediction observation get a higher weight in the decision process, when a kernel other then the *rectangular* is chosen. 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 kernels do not have tunable hyperparameters.

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. The option to apply a custom link function 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 statistical programming language R (R 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). The *mlr* package (Bischl et al., 2016) was used 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.



(a)

(b)

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

4. Results

4.1. Tuning and runtime

While ten (or more) hyperparameter tuning iterations substantially improved the performance of BRT and SVM classifiers compared to default hyperparameter values, KNN and RF hyperparameter tuning did not result in relevant changes in AUROC (Figure 3a). Fifty tuning iterations and more further improved BRT and SVM performances only slightly. For RF, all tested tuning iterations showed a small decrease in AUROC compared to the default values (*sp/not*). BRT showed the highest tuning effect of all models with an increase of ~0.08 AUROC (Figure 3a).

Notable differences between the spatial (*sp/sp*) and non-spatial (*sp/nsp*, *nsp/nsp*) tuning settings can be seen for RF and SVM when looking at the chosen optimal hyperparameter settings (Figure 4). For setting *sp/sp* RF hyperparameter m_{try}, which specifies the number of variables used at each split, values from 1 - 3 were most often among the winning setting with $m_{try} = 3$ being the setting that was chosen most often. In contrast, setting *sp/nsp* and *nsp/nsp* mainly favored $2 <= m_{try} <= 4$ and did not select $m_{try} = 1$ once.

SVM with RBF kernel reveals two strong linear patterns between optimal hyperparameters C and σ (Figure 4) for setting *sp/sp*. If the Cost parameter C increases, bandwith σ either stays at a value between 2^{-1} to 2^{-2} or decreases linearly towards the set parameter limit of 2^{-15}. In contrast, setting *nsp/nsp* and *sp/nsp* mainly cluster around $\sigma = 2^{-3}$ and $C = 2^1$. KNN in setting *sp/sp* mainly used $k > 200$ while in settings *sp/nsp* and *nsp/nsp* k was favored within a range between 50 - 100 in combination with higher values for hyperparameter *distance*.

Table 2: Repetition mean AUROC values (bold) and runtime (minutes) for each model and CV setting for 200 random search iterations.

	nsp/nsp	*nsp/nsp*	*sp/sp*	*sp/not*
SVM	**0.906**, 134.30	**0.858**, 152.16	**0.844**, 128.10	**0.814**, 0.31
RF	**0.945**, 1697.41	**0.872**, 1651.95	**0.863**, 1594.90	**0.872**, 0.28
BRT	**0.949**, 2923.66	**0.873**, 2905.61	**0.859**, 2895.17	**0.771**, 0.20
KNN	**0.895**, 86.67	**0.839**, 113.21	**0.839**, 81.66	**0.841**, 0.22

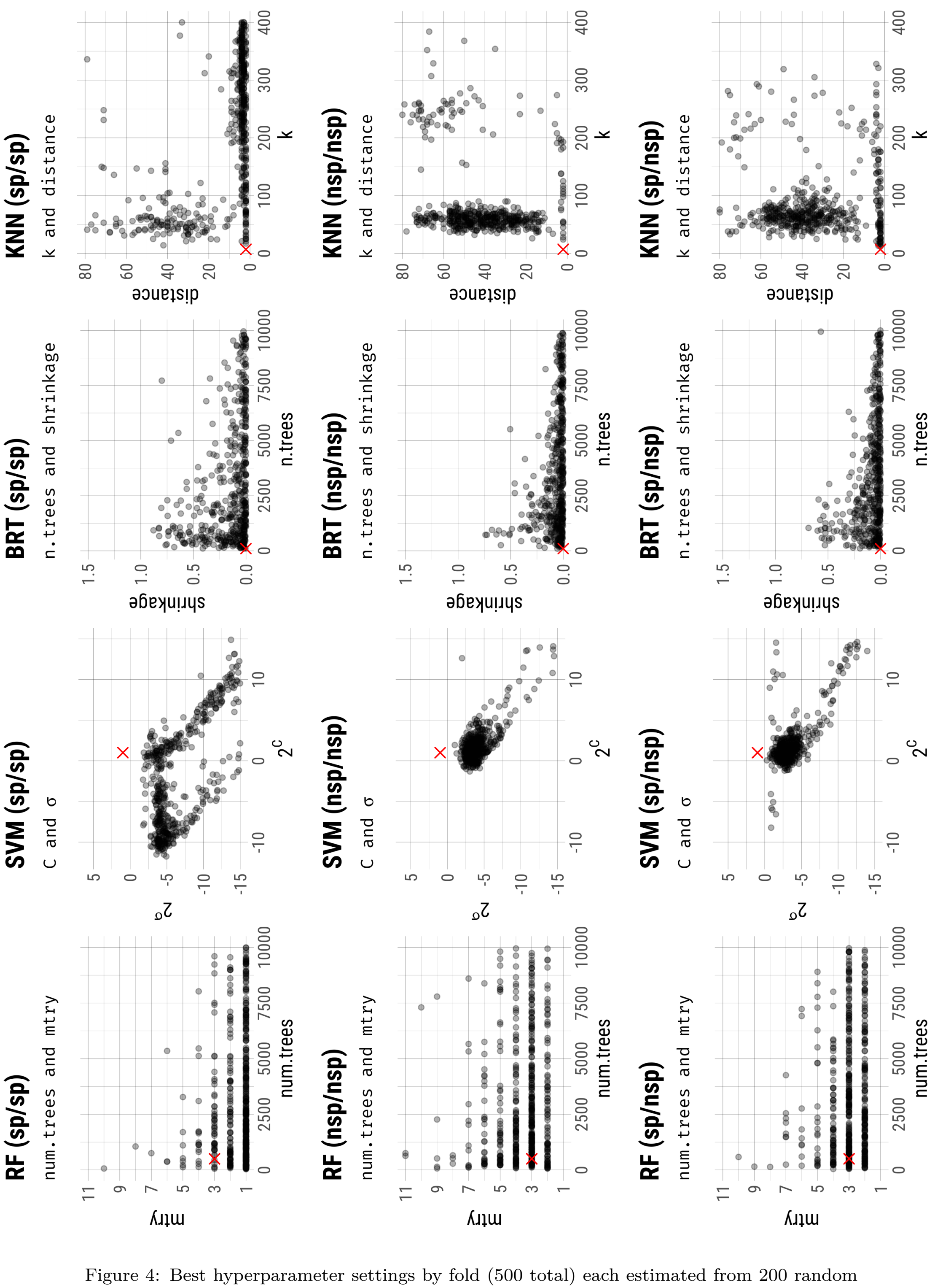


Figure 4: Best hyperparameter settings by fold (500 total) each estimated from 200 random search tuning iterations per fold using five-fold cross-validation. Split by spatial and non-spatial partitioning setup and model type. Red crosses indicate default hyperparameter values of the respective model. Black dots represent the winning hyperparameter setting out of each random search tuning of the respective fold.

4.2. Predictive performance

BRT shows the best predictive performance in the *nsp/nsp* setup but also the worst performance for the *sp/not* setting (Figure 5).

All models show overoptimistic performances for setting *nsp/nsp* due to spatial autocorrelation with GAM, GLM and BRT being the models profiting most (Figure 5). Parametric models (GAM, GLM) show an overall lower predictive performance between 0.05 - 0.1 AUROC compared to all non-parametric models considering the *sp/sp* setting.

RF and BRT showed roughly equally good predictive performances in setting *sp/sp* with only minor differences (RF shows a slightly higher mean (0.863 vs 0.859 AUROC) and median value (0.862 vs 0.861 AUROC) than BRT) (Figure 5). RF shows a small decrease in predictive performance for setting *sp/sp* compared to *sp/not* that is further analyzed in the discussion section.

5. Discussion

5.1. Tuning

Hyperparameter tuning is a tradeoff between number of iterations and runtime. The goal is to use as few tuning iterations as possible to find the best hyperparameter setting of a model for a specific data set. If the tuning dimension of a hyperparameter search space 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 settings are uniformly distributed over the search space. The higher the number of tuning iterations, the more likely it becomes that the resulting best hyperparameter setting is close to the theoretical optimum. However, it can not be verified if the optimum is found unless all possible combinations have been checked. This is impossible for a numeric search space and most of the time impracticable for a nominal or integer

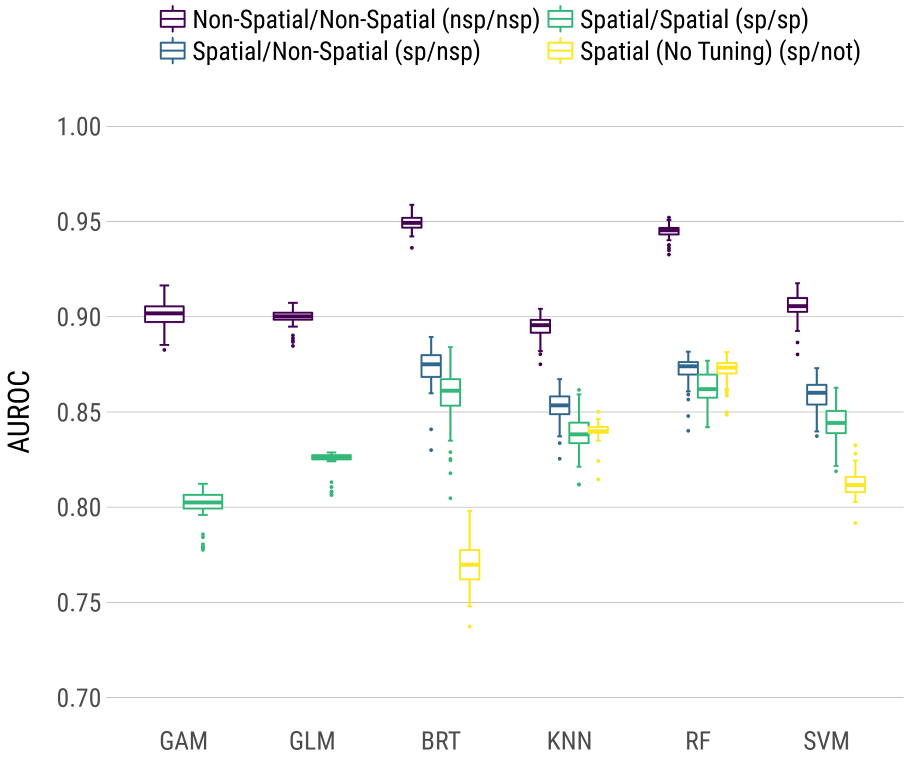


Figure 5: (Nested) CV estimates of model performance at the repetition level using 1000 random search iterations. CV setting refers to outer/inner loop of the respective (nested) CV, e.g. "Spatial/Non-Spatial" means that spatial partitioning was used in the outer loop and non-spatial partitioning in the inner loop. For GAM and GLM, only the outer loop setting applies as no tuning was performed.

search space. Bergstra & Bengio (2012) demonstrated that random search outperforms grid search in both runtime and predictive accuracy. Besides these two approaches, Bayesian Optimization and 'F-racing' are widely used for optimization of black-box models (Brochu et al., 2010; Malkomes et al., 2016; Birattari et al., 2002).

Depending on the data set characteristics, some models (e.g. RF) can be insensitive to hyperparameter tuning (Biau & Scornet, 2016; Díaz-Uriarte &

De Andres, 2006). As the effect of hyperparameter tuning always depends on the data set characteristics, we recommend to always perform a tuning of hyperparameters (e.g. 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 plausible parameters for each model. It should be ensured by visual inspection that the main portion of the optimum hyperparameter combinations of each fold does not hit the borders of the tuning space. If the optimal hyperparameter settings are clustered at the edge of the parameter limits, this implies that the model would possibly favorite hyperparameter values which lie outside the given range. However, extending the tuning space is not always possible nor practical as numerical problems within the algorithm may occur that may prohibit further extension of the tuning space. This especially applies models with a numerical search space (e.g. SVM) that increase exponentially. In a practical sense the user has to question himself if extending the parameter ranges could possibly result in a significant performance increase and is worth the tradeoff of having an increased runtime. All these points exacerbate the specification of parameter limits for hyperparameter tuning. As the optimal parameter limits also depend on the dataset characteristics, it is not possible to define an optimal search space for an algorithm upfront. The chosen parameter limits of this work can serve as a starting point for future analysis but do not claim to be optimal. Users should analysis parameter search spaces of various studies to find suitable limits that match their dataset characteristics. Within the framework of the *mlr* project a database exists which stores tuning setups of various models from users that can serve as a reference point (Richter et al., 2017).

Although there is a clear increase in performance if non-spatial tuning is used (*nsp/nsp*) compared to spatial tuning (*sp/nsp*), we recommend to use setting (*sp/sp*). We briefly outline the reasons: Setting *sp/nsp* causes overoptimistic performance estimates in the tuning loop as the algorithm can afford to adapt highly to the training data to get good performance results as the test data is relatively similar to the training data. Although these overoptimistic performance values in the tuning step do not directly alter the performance estimation in the outer loop, they are based on models which are to some degree highly adapted/overfitted to the training data. This is caused by the spatial autocorrelation (similarity of train - and test set) that applies if a non-spatial tuning setting is used. Such highly adapted models are based on hyperparameters that allow such a high adaption to the training data in the first place. Generally spoken, hyperparameters from a non-spatial tuning lead to more adapted models than hyperparameters estimated from a spatial tuning. Models fitted with hyperparameters from a non-spatial tuning can then profit from the remaining spatial autocorrelation in the train/test split in the outer loop because spatial partitioning is only capable of reducing spatial autocorrelation but cannot completely remove it. Hence, performance estimates of setting *sp/nsp* are somewhat more overoptimistic (biased) than setting *sp/sp*. However, the practical difference between *sp/nsp* and *sp/sp* for our dataset is very small, e.g. 0.01 AUROC for RF (0.872 vs. 0.862) or even 0.005 AUROC for BRT (0.854 vs. 0.859) (Table 2).

[COMMENT: Ist diese Hypothese mit RF im folgendem Absatz so tragbar bzw. kommunizierbar? Ich weiß, das ist eine heiße These. Und ja, ich verallgemeinere gerne.. aber die hyperparameter sind ja in der Tat nicht auf spatial data sets estimated worden back in time, also wäre es ja nicht wirklich abwegig, dass sie nicht optimal sind. Daher kann man diese These, basierend auf den Results hier, ja mal vorsichtig nennen? :) Bin gespannt auf deine Meinung.] It seems unexpected at a first glance that RF does even show a drop in performance when being spatially tuned (*sp/sp*) compared to no hyperparameter tuning (0.862 AUROC vs. 0.872 AUROC). This behaviour can to some degree be explained by the just explained behaviour of hyperparameter selection using spatial tuning and observed via the winning hyperparameter settings per fold (Figure 4): For setting *sp/sp*, winning *mtry* values are mainly 1 or 2 in our case. Low *mtry* values lead to more generalized models which do not that much adapt to the data than models trained with higher *mtry* values REFERENZ. The more a model adapts to the data, the better it can make use of spatial autocorrelation within training and test set. This is backed up by the winning hyperparameter settings of tuning settings *sp/nsp* and *nsp/nsp* which show, on average, higher *mtry* values than setting *sp/sp*. In both cases, the main portion of the winning *mtry* values for *sp/nsp* and *nsp/nsp* ranges between 2 - 4 while $mtry = 1$ is not present at all. Since the default value of *mtry* for classification cases is $\sqrt{n_{variables}}$ (rounded down) which resolves to $\sqrt{11} = 3$ in our case, this hyperparameter value leads to models that are more adapted to the data than most of the models fitted with the selected *mtry* values from the spatial tuning setting *sp/sp*, which mainly show values of $mtry < 3$. Subsequently, one could state that, for our dataset, the default *mtry* setting of RF creates models that result in somewhat overoptimistic performance estimates. As the default hyperparameter values of RF were initially determined by using various non-spatial datasets Breiman (2001), it could possibly be that this default value is not optimal for spatial datasets in general. However, this hypothesis must be analysed more closely using different spatial datasets and goes beyond the scope of this work.

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 hyperparameter tuning on the predictive performance varies. Although no significant increase or even small decreases in predictive accuracy may occur (e.g. for RF in this study), the user has to tune hyperparameters in any case as default values may not be meaningful (e.g. SVM, BRT) or eventually even cause overoptimistic models (e.g. RF).

5.2. Predictive Performance

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

The higher predictive performance of RF and BRT compared to all other models when looking at the spatial *sp/sp* marks these models as the winners 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 parametric 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 better performance of the GLM compared to the GAM suggests that generalized models show better predictive performance abilities on the dataset of this study. We did not perform stepwise variable selection or similar on the parameteric models (GLM, GAM) as we wanted to ensure that all models use the same predictor set. An exploratory analysis was done on using different starting basis dimensions for the optimal smoothing estimation of each predictor of the GAM. The reported GAM model was initiated with $k = 10$ as the basis dimension which ensured full flexibility of the smoothing terms for each predictor. Although RF and BRT showed the best predictive performance in our case, models like SVM and KNN 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, the main results of this study would look as follows: (i) The winning model would have been BRT only instead of RF and BRT. (ii) The predictive performance would been reported with a mean value of 0.949 AUROC which is ~0.087 AUROC higher than the bias-reduced performance estimated by spatial CV (*sp/sp*) (0.862 AUROC). 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.

Although the best trade-off is achieved by RF without hyperparameter tuning in our case, model interpretability is often an important point in ecological modeling to favor parametric models over non-parametric ones. If only a minor difference in performance exists, the user might think about choosing e.g. the GLM over RF for reasons of runtime and interpretability.

Another possible model selection criteria within the spatial modeling field is the quality of the prediction surface of a prediction map. However, this point is not analysed in this study as the focus is on hyperparameter tuning 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 surfaces. 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

If coefficients of parametric 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) 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 focused on predictive accuracy in this work, we did not use spatial autocorrelation structures during model fitting for GLM and GAM to reduce runtime.

Interpretability is an important attribute of an algorithm, if not even the most important one in ecological modeling. Ecologists often favor parametric 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., 2014). The latter are able to provide relative estimates of variable importance but do not provide coefficients to interpret the relationships between predictors and the 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 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, 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 parametric models in terms of predictive accuracy with RF and BRT showing the highest values. 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 for both hyperparameter tuning and performance estimation. Furthermore, we recommend to be clear on the analysis aim before conducting spatial modeling: If the goal is to understand environmental processes by statistical inference, parametric 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. We hope that this work helps in performing fair bias-reduced model performance comparisons that account for spatial data.

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 n_{trees} is the benchmark measure, respectively, compared to package *randomForest* (Wright & Ziegler, 2017). 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 *e*1071 (Meyer et al., 2017) package because *kernlab* offers more kernel options. Other kernels than RBF have been exploratively modeled but not analysed in detail in this work.

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

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 chosen 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	$\mathbf{q_1}$	$\widetilde{\mathbf{x}}$	$\bar{\mathbf{x}}$	$\mathbf{q_3}$	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	0.0	0.0	0.0	0.0	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.3: Descriptive summary statistics of numerical variables. Precipitation (p_sum) in mm/m^2, temperature (temp) in °C, solar radiation (r_sum) in kW/m^2, tree age (age) in years. Statistics show sample size (**n**), minimum (**Min**), 25% quantile ($\mathbf{q_1}$), median ($\widetilde{\mathbf{x}}$), mean ($\bar{\mathbf{x}}$), 75% quantile ($\mathbf{q_3}$), maximum (**Max**), inner-quartile range (**IQR**) and NA Count (**#NA**).

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	surface deposits	32	3.4
	clastic sedimentary rock	607	64.3
	biological sedimentary rock	141	14.9
	chemical sedimentary rock	151	16.0
	magmatic rock	13	1.4
	all	944	100.0
soil	young soils with small soil horizon difference	676	71.6
	soil with accumulation of organic material	25	2.6
	limited space for roots	19	2.0
	soil with accumulation of nitrates	13	1.4
	soil influenced by ferric or similar	18	1.9
	water influenced soil	21	2.2
	organic soil	15	1.6
	soil with clay in subsoil	157	16.6
	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

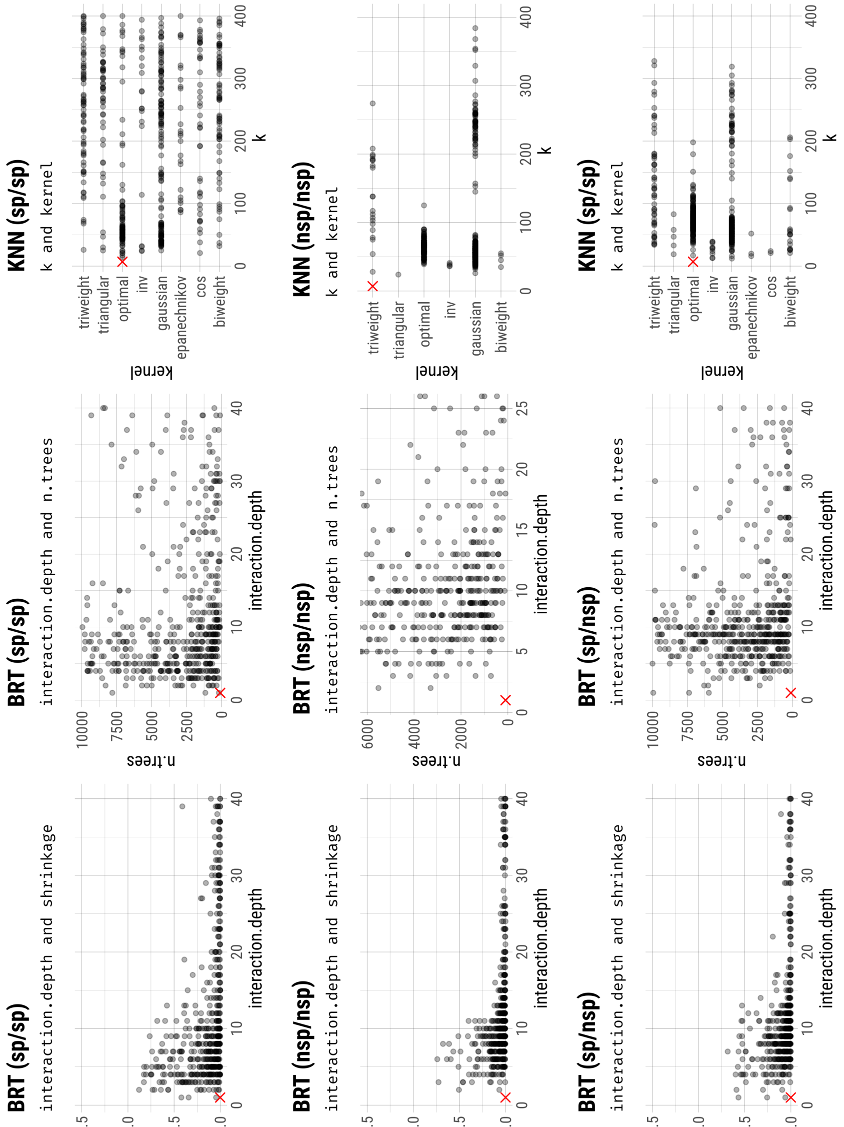


Figure C.6: Best hyperparameter settings by fold (500 total) each estimated from 200 random search tuning iterations per fold using five-fold cross-validation. Split by spatial and non-spatial partitioning setup and model type. Red crosses indicate default hyperparameter values of the respective model. Black dots represent the winning hyperparameter setting 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`.

Birattari, M., Stützle, T., Paquete, L., & Varrentrapp, K. (2002). A racing algorithm for configuring metaheuristics. In *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation* (pp. 11–18). Morgan Kaufmann Publishers Inc.

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 assessment of prediction rules in remote sensing: The R package sperrorest. In *2012 IEEE International Geoscience and Remote Sensing Symposium*. 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-Páez, A. P., & Muenchow, J. (2015). Landslide susceptibility near highways is increased by 1 order of magnitude in the Andes of southern Ecuador, Loja province. *Natural Hazards and Earth System Sciences*, *15*, 45–57. URL: `http://www.nat-hazards-earth-syst-sci.net/15/45/2015/`.

Brochu, E., Cora, V. M., & de Freitas, N. (2010). A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling 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 assessment of the efficacy of support vector machines, artificial neural networks, 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 (ROC) Curves: An Analysis Tool for Detection Performance*. Technical Report. 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.

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. 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.

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 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: 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. 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., & Iturritxa, E. (2009). A global climatic risk assessment of pitch canker disease. *Canadian Journal of Forest Research*, *39*, 2246–2256. URL: `https://doi.org/10.1139%2Fx09-131`. 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 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: Challenges, methods, and tools. *Proceedings of the International Astronomical Union*, *12*, 345–348. doi:`10.1017/S1743921316012813`.

Geiß, C., Pelizari, P. A., Schrade, H., Brenning, A., & Taubenböck, 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 Geoscience and Remote Sensing Letters*, *14*, 2008–2012. doi:`10.1109/LGRS.2017.2747222`.

GeoEuskadi (1999). *Litologia 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 Publishing. URL: `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 models. *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.

Guo, Q., Kelly, M., & Graham, C. H. (2005). Support vector machines for predicting distribution of sudden oak death in california. *Ecological Modelling*, *182*, 75–90. URL: https://doi.org/10.1016%2Fj.ecolmodel.2004.07.012. doi:10.1016/j.ecolmodel.2004.07.012.

Halvorsen, R., Mazzoni, S., Dirksen, J. W., Næsset, E., Gobakken, T., & Ohlson, M. (2016). How important are choice of model selection method and spatial autocorrelation of presence data for distribution modelling by MaxEnt? *Ecological Modelling*, *328*, 108–118. URL: https://doi.org/10.1016%2Fj.ecolmodel.2016.02.021. doi:10.1016/j.ecolmodel.2016.02.021.

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.

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). SoilGrids250m: Global gridded soil information based on machine learning. *PLOS ONE*, *12*, e0169748. URL: https://doi.org/10.1371%2Fjournal.pone.0169748. doi:10.1371/journal.pone.0169748.

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%2Fj.1365-3059.2010.02380.x. doi:10.1111/j.1365-3059.2010.02380.x.

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%2Fs12665-015-4866-9. doi:10.1007/s12665-015-4866-9.

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%2Fssci.2016.7850221. doi:10.1109/ssci.2016.7850221.

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%2F978-3-642-25566-3_40. doi:10.1007/978-3-642-25566-3_40.

Iturritxa, E., Mesanza, N., & Brenning, A. (2014). Spatial analysis of the risk 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%2F978-1-4614-7138-7. 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 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.

Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai* (pp. 1137–1145). Stanford, CA volume 14.

Legendre, P., & Fortin, M. J. (1989). Spatial pattern and ecological analysis. *Vegetatio*, *80*, 107–138. URL: https://doi.org/10.1007%2Fbf00048036. doi:10.1007/bf00048036.

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.

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.

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.

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.

Muenchow, J., Feilhauer, H., Bräuning, A., Rodríguez, E. F., Bayer, F., Rodríguez, R. A., & Wehrden, H. (2013a). Coupling ordination techniques and 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.

Muenchow, J., Hauenstein, S., Bräuning, A., Bäumler, R., Rodríguez, E. F., & von Wehrden, H. (2013b). 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 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 â€“ case study 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.

Pradhan, B. (2013). A comparative study on the predictive ability of the decision tree, support vector machine and neuro-fuzzy models in landslide susceptibility mapping using GIS. *Computers & Geosciences*, *51*, 350–365. URL: https://doi.org/10.1016%2Fj.cageo.2012.08.023. doi:10.1016/j.cageo.2012.08.023.

Quillfeldt, P., Engler, J. O., Silk, J. R., & Phillips, R. A. (2017). Influence of device accuracy and choice of algorithm for species distribution modelling of seabirds: a case study using black-browed albatrosses. *Journal of Avian Biology*, . URL: https://doi.org/10.1111%2Fjav.01238. doi:10.1111/jav.01238.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing Vienna, Austria. URL: https://www.R-project.org/ R version 3.3.3.

Richter, J., your git settings!, C., Bischl, B., & mmomIvan (2017). jakob-r/mlrhyperopt: First beta. URL: https://doi.org/10.5281/zenodo.896269. doi:10.5281/zenodo.896269.

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. 184–195). Springer Berlin Heidelberg. URL: https://doi.org/10.1007%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 example from precision agriculture. In *Advances in Data Mining. Applications and Theoretical Aspects* (pp. 450–463). Springer Berlin Heidelberg. URL: 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). Spatial modeling and geovisualization of rental prices for real estate portals. *International Journal of Agricultural and Environmental Information Systems*, *8*, 78–91. URL: https://doi.org/10.4018%2Fijaeis.2017040106. doi:10.4018/ijaeis.2017040106.

Schliep, K., & Hechenbichler, K. (2016). *kknn: Weighted k-Nearest Neighbors*. URL: https://CRAN.R-project.org/package=kknn R package version 1.3.1.

Smoliński, S., & Radtke, K. (2016). Spatial prediction of demersal fish diversity in the Baltic Sea: Comparison of machine learning and regression-based techniques. *ICES Journal of Marine Science: Journal du Conseil*, (p. fsw136). URL: https://doi.org/10.1093%2Ficesjms%2Ffsw136. doi:10.1093/icesjms/fsw136.

Stelmaszczuk-Górska, M., Thiel, C., & Schmullius, C. (2017). Remote sensing for aboveground biomass estimation in boreal forests. In *Earth Observation for Land and Emergency Monitoring* (pp. 33–55). John Wiley & 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 *Nonlinear Modeling* (pp. 55–85). Springer US. URL: https://doi.org/10.1007%2F978-1-4615-5703-6_3. doi:10.1007/978-1-4615-5703-6_3.

Vorpahl, P., Elsenbeer, H., Märker, M., & Schröder, B. (2012). How can statistical models help to determine driving factors of landslides? *Ecological Modelling*, *239*, 27–39. URL: https://doi.org/10.1016%2Fj.ecolmodel.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 potential geographic distribution of Radopholus similis in China. *Agricultural 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*, 723–735. URL: https://doi.org/10.1007%2Fs10530-006-9072-y. doi:10.1007/s10530-006-9072-y.

Wieland, R., Kerkow, A., Früh, 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.

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.

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`.

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`.

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`.

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 York. URL: `https://doi.org/10.1007%2F978-0-387-87458-6`. doi:`10.1007/978-0-387-87458-6`.