RUSBoost: A suitable species distribution method for imbalanced records of presence and absence. A case study of twenty-five species of Iberian bats

Jaime Carrasco^{a,b}, Fulgencio Lisón^{c,d} and Andrés Weintraub^{a,b}

October 7, 2021

^aUniversity of Chile, Industrial Engineering Department, Santiago, Chile.

^bComplex Engineering System Institute - ISCI, Santiago, Chile.

^cLaboratorio de Ecología del Paisaje y Conservación, Departamento de Ciencias Forestales y Medioambiente, Facultad de Ciencias Agropecuarias y Forestales, Universidad de La Frontera, Box 54-D, Temuco, Chile.

^dWildlife Ecology and Conservation Lab, Departamento de Zoología, Facultad de Ciencias Naturales y Oceanográficas, Universidad de Concepción, Casilla 160-C, Concepción, Chile

Abstract

1. Traditional Species Distribution Models (SDMs) may not be appropriate when examples of one class (e.g. absence or pseudo-absences) greatly outnumber examples of the other class (e.g. presences or observations), because they tend to favor the learning of observations more frequently.
2. We present an ensemble method called **Random UnderSampling and Boosting** (RUSBoost), which was designed to address the case where the number of presence and absence records are imbalanced, and we opened the “black-box” of the algorithm to interpret its results and applicability in ecology.
3. We applied our methodology to a case study of twenty-five species of bats from the Iberian Peninsula and we build a RUSBoost model for each species. Furthermore, in order to improve to build tighter models, we optimized their hyperparameters using Bayesian Optimization. In particular, we implemented a objective function that represents the cross-validation loss: $kFoldLoss(\vec{z})$, with $\vec{z}$ representing the hyper-parameters *Maximum Number of Splits*, *Number of Learners* and *Learning Rate*.
4. The models reached average values for Area Under the ROC Curve (AUC), specificity, sensitivity, and overall accuracy of $0.84 \pm 0.05\%$, $79.5 \pm 4.87\%$, $74.9 \pm 6.05\%$, and $78.8 \pm 5.0\%$, respectively. We also obtained values of variable importance and we analyzed the relationships between explanatory variables and bat presence probability.
5. The results of our study showed that RUSBoost could be a useful tool to develop SDMs with good performance when the presence/absence databases are imbalanced. The application of this algorithm could improve the prediction of SDMs and help in conservation biology and management.

Keywords: bats; habitat suitable models; imbalanced presence-absence data; species distribution modeling.

1 Introduction

Artificial intelligence methods, particularly those of Machine Learning (ML), are useful in finding complex interactions between the studied variables directly from the data (Flach, 2001; Olden et al., 2008), and have been recognized as formidable scientific tool in natural resource management and wildlife biology (Elith and Leathwick, 2009; Olden et al., 2008). ML approach can be applied to virtually any data and eliminates the need to specify, *a priori*, broadly untested and potentially biased assumptions regarding the underlying statistical distribution of the data (Breiman et al., 2001). Moreover, various studies have indicated that traditional statistical methods are not suitable for problems that come from ecology or related fields due to their low predictive power (Breiman et al., 2001; Elith et al., 2008).

Most of the tools for modeling species distribution today come from ML methods. In particular, supervised learning models are closely related to Species Distribution Models (SDMs), which attempts to predict the probability of the presence of a species, given certain variables that characterize the underlying environment (Wisz et al., 2013; Jiménez-Valverde et al., 2013; Thibaud et al., 2014; García-Callejas and Araújo, 2016). In the last decade, several methods such as Classification and Regression Trees (Breiman et al., 2017), Random Forest (Breiman, 2001), Boosted Trees (Freund et al., 1996), Support Vector Machines (Schölkopf et al., 2001), Artificial Neural Networks (Bishop et al., 1995; Goodfellow et al., 2016) and so forth, have been intensively used in different applications that involve the modeling of the species distribution, or also commonly referred to as ecological niche models (ENMs) (Cutler et al., 2007; De'Ath, 2007; Drake et al., 2006; Lek and Guégan, 2012; Olden et al., 2008; Phillips et al., 2006). In general terms, the purpose of the SDM is to predict the spatial variation of the presence or abundance of species within the occupied environmental space. For this purpose, biotic or abiotic variables/drivers are constructed, usually associated with the geographical place where the species has been observed, and depending on the modeling approach, variables are also assigned to the places where they have not been documented (Elith and Leathwick, 2009; Pulliam, 2000; Soberón and Nakamura, 2009). Based on the information provided in the modeling process, SDMs can be divided into two branches: i) those that are based on data that contain only presence (Elith et al., 2006; Thibaud et al., 2014; Wisz et al., 2013; Tsoar et al., 2007) or, ii) those that record the presences and absences (De'Ath, 2007; Cutler et al., 2007). In ML, the second branch is a typical two-class classification problem, or also called a binary classification (presence and absence classes). In contrast, the former has been developed mainly using methods that do not come directly from the ML approach, for example Convex Hull approach (Cornwell et al., 2006) or BIOCLIM using depth functions (Busby, 1991), to name a few examples.

There is still a debate about which of the two approaches is the most appropriate. Some comparative studies indicated that the models that used presence and absence data were superior to those that only used presence (Brotons et al., 2004; Elith et al., 2006). Others emphasize presence-only models for conceptual reasons, stating that one-class models more elegantly represent the idea of niche rather two-class models (Drake et al., 2006; Drake and Bossenbroek, 2009); and some recommend one or the other approach depending: if the absence data is reliable, or if it is necessary to optimize the sensitivity or specificity metrics (Maher et al., 2014). However, what is not up for discussion is the balancing of data in the presence/absence approach: whether certain classes, for example the presence observations are underrepresented in the dataset, traditional models tend to favor the learning of observations more frequently, producing a classifier biased towards the majority class (Richhariya and Tanveer, 2020; Seiffert et al., 2009). For example, when fauna is modeled, it is common that this phenomenon is likely to occur, due to the biological and behavioral characteristics of the animals (elusive animals, misidentification in the capture processes, incomplete or biased monitoring, inaccessible areas, rare species, etc.) (Soley-Guardia et al. (2016); Sillero et al. (2021); Velasco and González-Salazar (2019)). Therefore, finding algorithms capable of improving predictions with this type of imbalance is important when it comes to providing useful tools for managing biodiversity conservation at different levels.

The problem of learning from imbalanced data has been carefully dealt in different ways and for

different methods. (Chawla et al., 2002) introduced a method that over-samples the minority class and under-samples the majority class called Synthetic Minority Oversampling Technique (SMOTE), and tested its solution with an algorithm used to generate a decision tree (C4.5) (Quinlan, 1992), a Rule-based classification algorithm called Ripper (Cohen, 1995) and a Naive Bayes classifiers (Mladenic and Grobelnik, 1999). Seiffert et al. (2009) combined Random UnderSampling (RUS) and Boosting (RUSBoost) (Freund et al., 1996) providing a simple and efficient method for improving predictive performances when the database containing the species’ presence and absence records were imbalanced. Recently, Richhariya and Tanveer (2020) uses oversampling and undersampling of data to remove the imbalance in the classes and “*universum*” learning with SVM method. Currently, these techniques and procedures that deal with the problem of data imbalance have been little explored in studies whose purpose is modeling the distribution of species and only there are few in the literature (Breiner et al. (2015), Johnson et al. (2012), and Robinson et al. (2018)). This is an important issue when the data comes from species which are problematic in their identification or which have not been subjected to an equal survey effort.

In our study, we hypothesized that the RUSBoost method is suitable for modeling the distribution of species that contain records of presence and absence data, and its performance is independent of the level of imbalance between presence and absence data. Therefore, our aims were: i) to explain in detail how the method works, in algorithmic and interpretability and applicability in ecological terms; and, ii) to apply our methodology to a case study of twenty-five species of bats from the Iberian Peninsula and analyze the results. In addition, we present the Bayesian Optimization procedure (Eggensperger et al., 2013) to find the optimal hyperparameters of the method, to achieve the best performance of the models (one for each species).

2 Materials and methods

2.1 Context, bat species and predictors

Bats are a diverse mammals group very threatened and they need conservation policies. However, bats are nocturnal animals and the different survey techniques could affect our capacity to detect some species. Therefore, it is usual to find distribution maps imbalanced for these mammals. For this reason, we used presence/absence distribution data of bat species present in the Iberian Peninsula. This is a biogeographical region with an area about six millions of square kilometers divided into $10 \times 10 \text{ km}^2$ UTM cells. The dataset was elaborated from the distribution maps of “Atlas y Libro Rojo de los Mamíferos Terrestres de España” (Palomo et al., 2007) and “Atlas dos morcegos de Portugal” (Rainho et al.); and was recently updated with new records with the same resolution from personal surveys and references (Lisón et al., 2015; Lisón and Sánchez-Fernández, 2017). We did not use those bat species where there were less than thirty records. Also, we did not include the twin species *Myotis nattereri/escalerai* because there is uncertainty with its distribution (Palomo et al., 2007; Juste et al., 2018). For the species *Eptesicus serotinus* and *E. isabellinus*, we considered the distribution of the last one as limited to south and south eastern of Iberia (Andalusia and Murcia) (Palomo et al., 2007; Lisón et al., 2015).

Land-use variables were obtained following the procedure by Lisón et al. (2015) and Lisón and Sánchez-Fernández (2017). We used the land-use map provided by the CORINE Land Cover Project (see www.eea.europa.eu). Also, we modelled the habitat suitable models of bat species using as environmental variables the land-use cover from CORINE Land Cover Project for 2006 (<https://land.copernicus.eu/pan-european/corine-land-cover/clc-2006>). The land-use cover was reclassified from the 44 initial categories to 16 final categories (see Lisón and Sánchez-Fernández (2017)). For each final category, we calculated the percentage of surface occupied in each cell. Additionally, we used 44 environmental and geographical variables extracted of BIOMOD for each cell following Barbosa et al. (2012) (Table 1).

Our database was characterized by the high degree of imbalance in the records of presence and

Drivers	Variable	Description	Variable	Description
Human Activity	u100	Distance to the nearest town with more than 100,000 inhabitants (km)	u500	Distance to the nearest town with more than 500,000 inhabitants (km)
	daut	Minimum distance road (km)	-	-
Geographic and Topographic	alti	Average altitude (m)	dalt	Difference in altitude (slope, m)
	pend	Average slope	lati	Latitude
	long	Longitude	-	-
Weather	tene	Average temperature in January	huen	Average humidity in January
	tjul	Average temperature in July	huju	Relative humidity in July
	tmed	Average annual temperature (°C)	vhum	Annual relative air humidity range
	vtem	Annual temperature range (°C) = (TJul-TJan)	perm	Soil Permeability
	inso	Mean annual insolation (hours/year)	rads	Mean annual sun radiation (Kwh/m ² /day)
	etp	Mean annual potential evapotranspiration (mm)	etr	Mean annual real evapotranspiration
	prec	Mean annual precipitation (mm)	dipr	Mean annual number of days with precipitation >= 0.1 mm
	pm24	Maximum precipitation in 24 hours	pmr	Relative maximum precipitation (=PM24/Pmed)
	dihe	Frost days	-	-
Land cover	Improductive	Improductive [%]	Urban	Urban areas [%]
	Roads	Road [%]	Mines	Mine zones [%]
	Rainfallcrops	Rainfall crops [%]	Irrigatedcrops	Irrigate crops [%]
	Pasture	Pasture [%]	Mixedcrops	Mixed crops [%]
	Abandoned	Abandoned crops [%]	Agroforestry	Agroforestry areas [%]
	Forest	Forest [%]	Shurbland	Shrubland
	SandRock	Sand and rock [%]	River	Rivers [%]
	Wetland	Wetland [%]	-	-

Table 1: Description of predictor variables.

absence among bat species and geographic range that the species cover. Fig. 1 shows that the number of records comprise a range from 33 to 2,297, with quartiles of $Q_1 = 212.5$ and $Q_2 = 522$ and $Q_3 = 914.25$. Furthermore, the records were very scattered covering a latitudinal range of 36.06° to 43.7° ($Q_1 = 39.42$, $Q_2 = 41.05$, $Q_3 = 42.21$) and longitudinal range of -9.47° to 3.29° ($Q_1 = -6.53$, $Q_2 = -4.46$, $Q_3 = -2.06$).

The basic notation and terminology that will be used throughout this section will be as follows: let $\vec{x} \in \mathbb{R}^d$ be the predictor variables described in Table 1 (then $d = 36$), and by $y \in \{0, 1\}$ response variable representing absence or presence of the bat species, labeled by 0 and 1 respectively. Let $\mathcal{I}$ be the cells index set of study area, and let $\mathcal{D} := \{(\vec{x}^i, y^i), i = 1, \dots, |\mathcal{I}|\}$ be the dataset, where let $\vec{x}^i$ be to the predictor vector labeled by $y^i \in \{0, 1\}$ and, $y^i = 1$ if it is a presence observation and $y^i = 0$ if it is an absence. When necessary, we will refer to y_b to the response variable corresponding

Figure 1: Bat records.

111 to species $b \in \mathcal{B}$, where $\mathcal{B} = \{1, \dots, 25\}$ (see Table S1). Notice that $R_b := \sum_{i \in \mathcal{I}} y_b^i$ represents the
 112 occurrence/presence number of bat species b in our dataset, and therefore the Fig. 1 (top) shows its
 113 boxplot.

114 2.2 How RUSBoost works

115 RUSBoost method combines Random UnderSampling (RUS) and AdaBoostM1 algorithm. Specif-
 116 ically, RUS is a technique which randomly removes data points from the majority class, and Ad-
 117 aBoostM1 is a boosting technique used as an Ensemble Method (Freund et al., 1996) (see Fig. 2)
 118 which sequentially trains a next learner model on the mistakes of the previous learners. In mathe-
 119 matical terms, RUS extracts random subsets $\mathcal{D}_l$ (with replacement) of dataset $\mathcal{D}$ of size $2 \cdot R < |\mathcal{I}|$,
 120 where R denotes the number of members in the class with the fewest members in the dataset. In
 121 our study, as the bat presence records are the minority class, then R will be set (for each model) to
 122 R_b . Subsequently, AdaBoostM1 builds a model (or learner), denoted $\mathcal{M}_l$, using $\mathcal{D}_l$ as sub-dataset (or
 123 training set), for $l = 1, \dots, L$, with L denoting the number of learners/sets. Later, classifiers $\mathcal{M}_l$ are
 124 used in parallel, each one offering an opinion $\mathcal{M}_l(\vec{x})$ as to which class the example should be labeled
 125 with, to predict the most likely class for an unobserved vector $\vec{x}$. A “master classifier” (denoted by
 126 $\mathcal{M}$, see Fig. 2) collects this information and then chooses the label y that has received more votes as
 127 follows:

$$y = \begin{cases} 1, & \tilde{y} > 0.5, \\ 0 & \text{otherwise,} \end{cases} \quad \text{where } \tilde{y} := \sum_{l=1}^L \alpha_l \mathcal{M}_l(\vec{x}) \quad \text{and} \quad \alpha_l = \log \frac{1 - \epsilon_l}{\epsilon_l}. \quad (1)$$

128 where the summation Eq. 1 can be seen as a weighted voting scheme and ϵ_l denotes the weighted
 129 classification error:

$$\epsilon_l = \sum_{i \in \mathcal{I}} w_l^i \cdot \mathbb{1}_{\{\neq 0\}}(\mathcal{M}_l(\vec{x}^i) - y^i), \quad (2)$$

130 and $\mathbb{1}_{\{\neq 0\}}(\cdot)$ is the indicator function. In our implementation, we normalize the α_l -parameters, so
 131 that $\sum_{l=1}^L \alpha_l = 1$, and we denoted $\tilde{y}$ by $\mathcal{P}(\vec{x})$, the probability of bat occurrence for the variable vector
 132 $\vec{x}$.

133 In Fig. 2, for $i \in \mathcal{I}$, the weight w_l^i represents the probability that the observation pair $(\vec{x}^i, y^i)$ is
 134 drawn to train the l -th learner model. AdaBoostM1 increases weights for observations misclassified
 135 by learner $\mathcal{M}_l$ and reduces (or keep them) weights for observations correctly classified. Then, next
 136 learner $l + 1$ is trained on the data with updated weights $(w_{l+1}^i)_{i \in \mathcal{I}}$. Models with $\epsilon_l \geq 1/2$ are
 137 discarded, and therefore the final model $\mathcal{M}$ is only assembled with models $\mathcal{M}_l$ that meet $\epsilon_l < 1/2$,

Figure 2: RUSBoost scheme: reading from top to bottom, the first L subsets of data are created from training sample $\mathcal{D}$ using RUS procedure. Then, each subset data $\mathcal{D}_l$ is used to train their decision trees $\mathcal{M}_l$ into the AdaBoostM1; and in this way, a set of different models are obtained sequentially. Finally, $(\mathcal{M}_l)_{l=1:L}$ models are combined into a “stronger ensemble” $\mathcal{M}$. The output response of $\mathcal{M}$ is calculated by the Eq. (1) (right side). Using the data set $\bar{\mathcal{D}}$ and Eq. (4), the accuracy level of $\mathcal{M}$ is calculated.

so that $(1 - \epsilon_l)/\epsilon_l > 1$, for all $l = 1, \dots, L$. From this, the procedure for updating the weights is as follows:

$$w_{l+1}^i = \begin{cases} \left(\frac{1-\epsilon_l}{\epsilon_l} \right) w_l^i, & \text{if } i \text{ is misclassified,} \\ 1 \cdot w_l^i, & \text{otherwise.} \end{cases} \quad (3)$$

2.2.1 Accuracy Assessment

After training the model, a validation accuracy score estimates the performance of the model on an independent data set $\bar{\mathcal{D}} := \{(\vec{x}^i, y^i), i = 1, \dots, |\bar{\mathcal{I}}|\}$. This procedure is known as *holdout validation*. It is worth mentioning that the full data set is such that $\tilde{\mathcal{D}} = \mathcal{D} \cup \bar{\mathcal{D}}, \mathcal{D} \cap \bar{\mathcal{D}} = \emptyset$; also that it is common to set $\mathcal{D}$ to 75% of the entire dataset and $\bar{\mathcal{D}}$ to 25%. Now, let $\mathcal{M}$ be the ensemble model built following the procedure described in the previous section, using the training set $\mathcal{D}$. Then, the accuracy of the method is defined by:

$$Accuracy(\mathcal{M} | \mathcal{D}, \bar{\mathcal{D}}) = \frac{1}{|\bar{\mathcal{I}}|} \sum_{i \in \bar{\mathcal{D}}} \mathbb{1}(y^i - \mathcal{M}(\vec{x}^i)), \quad (4)$$

147 $Accuracy(\mathcal{M} | \mathcal{D}, \bar{\mathcal{D}})$ represents the percentage of successes obtained by the model on the validation
 148 dataset $\bar{\mathcal{D}}$. When the datasets of observations are divided into k disjoint subsamples (or folds), chosen
 149 randomly but with roughly equal size; then is taken a group as a hold out or test dataset ($\bar{\mathcal{D}}$) and
 150 the remaining groups as a training dataset ($\mathcal{D}$), this procedure is known as *K-Fold Cross-Validation*.
 151 In our study, we adopted the latter procedure (with $K = 5$) to validate and protect our model from
 152 overfitting and we estimate the average classification error as:

$$kFoldLoss = 1 - \frac{1}{K} \sum_{k=1}^K Accuracy(\mathcal{M}^k | \mathcal{D}^k, \bar{\mathcal{D}}^k). \quad (5)$$

153 In addition, *specificity* and *sensitivity* are employed to appraise the classification capability com-
 154 puted in the following manner:

$$specificity = 100 \times \frac{TN}{FP + TN}, \quad sensitivity = 100 \times \frac{TP}{TP + FN}, \quad (6)$$

155 where TP (true positive) and TN (true negative) are the number of samples that are correctly clas-
 156 sified as positive (presence class) and negative (absence class) observations in the cross validation
 157 process, respectively. FP (false positive) and FN (false negative) are the number of samples that are
 158 misclassified. Therefore, *sensitivity* is the percentage of positive (presence class) observations that are
 159 correctly classified whereas *specificity* is the percentage of negative (absence class) observations that
 160 are correctly identified. In addition, an area under the curve (AUC) of a receiver operating charac-
 161 teristic (ROC) curve was used for evaluating the averaged performances of classifiers (Breiman et al.,
 162 2001), which shows the true positive (TP) rate versus the false positive (FP) rate attained by the
 163 model for different thresholds of the classifier output. This way, we can monitor if the model is able
 164 to extract and learn the characteristics that distinguish bat presence from absences ones. The ROC
 165 chart for a good classifier tend to rise sharply at the origin and then level off near the maximum value
 166 of 1.0. Finally, we consider Area Under the ROC Curve (AUC), this metric is generally considered
 167 to be an important index to quantitatively assess the overall accuracy of the classifier’s performance.
 168 An AUC value near 0.5 means that the predictive ability of the model is completely random and a
 169 value of 1.0 represents a perfect prediction without misclassification. The closer the AUC value is to
 170 1, the better the performance of the bat binary occurrence model.

171 2.2.2 Hyper-parameter optimization

172 The performance of machine learning algorithms, broadly depend on parameters (named hyper-
 173 parameters) that are calibrated by the user’s expertise (which is difficult and time-consuming), rec-
 174 ommendations from the algorithm package, or by more sophisticated methods that find the “best
 175 parameters” which improve some metric performance of the method (Yang and Shami, 2020). The
 176 hyper-parameters are internal parameters of any classifier or regression model and has a direct im-
 177 pact on the model’s performance. In this study, we used a Bayesian Optimization Method (BOM)
 178 (Eggenberger et al., 2013) to automate hyper-parameter settings used in the RUSBoost method
 179 (practical examples in Snoek et al. (2012)). In particular, we optimized the “Maximum Number of
 180 Splits (MNS)”, the “Number of Learners (NL)”, and the “Learning Rate (LR)” hyper-parameters.
 181 Higher MNS values lead to wider and more detailed decision trees as the number of leaves is in-
 182 creased, being able to identify more accurate split thresholds for the features. However, larger values
 183 may cause overfitting of the model to the training data in order to maximize its accuracy. Increasing
 184 the number of learners NL leads to a more time-consuming but potentially more accurate training
 185 of the model as the number of independent models learning from the data is increased. This allows
 186 the model to explore alternative classification schemes and focus on those difficult cases to adjust the
 187 weights of incorrectly classified instances (see α_i in the Eq. (1)). However, increasing the number of
 188 learners does not guarantee better performance (a priori). A natural trade-off between NL and LR

arises as smaller learning rates lead to “slow down” the learning in the model, decreasing the contribution of each classifier. This plays a crucial role to avoid overfitting to the training data (a common drawback of decision tree based models), obtaining a more general model. On the other hand, larger LR values lead to faster convergence, potentially with the same or better performance than smaller rates. Therefore, LR acts as a weighting factor for how quickly the model learns from the classifiers, existing an inherent trade-off between these two parameters. Given the highly combinatorial nature of the problem, we require an efficient method to explore and evaluate promising combinations of these crucial hyper-parameters in practical running times.

Broadly, BOM attempts to minimize (or maximize) a scalar objective function $F(\vec{\theta})$ for $\vec{\theta}$ in a bounded domain Θ where lies of hyperparameters. No additional assumptions are made about F , and then F can be seen as a black-box function. In our study, the function $F(\vec{\theta})$ represents the cross-validation loss $kFoldLoss(\vec{\theta})$ (see Eq. (5)) of the model (we set $K = 5$) and variables $\vec{\theta}$ its hyper-parameters, and therefore, minimizing $kFoldLoss$ function implies improving the predictive power of the model of both classes. The components/variables of $\vec{\theta}$ can be continuous reals, integers, or categorical. In our implementation let θ_{MNS} and θ_{NL} be the MNS and NL integer bounded variables; and θ_{LR} a LR real continuous bound variable. In particular, we arbitrarily assume that $10 \leq \theta_{MNS} \leq 500$, $1 \leq \theta_{NL} \leq |\mathcal{D}| - 1$ and $10^{-3} \leq \theta_{LR} \leq 1.0$. Thus, $\Theta = [10, 500] \times [1, |\mathcal{D}| - 1] \times [10^{-3}, 1] \subset \mathbb{Z}_+ \times \mathbb{Z}_+ \times \mathbb{R}_+$. Therefore, the optimization problem solved is:

$$\min kFoldLoss(\vec{\theta}), \quad \text{subject to } \vec{\theta} \in \Theta \quad (7)$$

and whose optimal values we will denote by $\vec{\theta}^* = (\theta_{MNS}^*, \theta_{NL}^*, \theta_{LR}^*)$.

2.3 Exploitation of RUSBoost for interpretability and decision making

A critical aspect of ML models is their interpretability (Castelvecchi, 2016). Despite the development of algorithms with remarkable performance, researchers and ecologists often fail to understand the logic behind the outputs; instead, they use them as an efficient black box. The previous sections are within this perspective, but we still need to understand what the role of variables is in predicting the probability of presence; or in the presence/absence classification process, and this is what we analyze in this section.

2.3.1 Variable importance measure

There are several ways to determine the relative importance of variables and it depends mainly on how the method works. Variable importance in ensemble methods is derived from the feature importance provided by its weak learner (or base classifier) (see $\mathcal{M}_l$ in the Fig. 2). In our approach we used a decision tree as a base classifier and then the AdaBoost feature importance was determined by the average feature importance provided by each decision tree.

“Predictor Importance (PI)” (Breiman, 1996) is a measure that assigns a score to each variable based on its contribution to the model learning process. With PI, candidate predictor variables can be compared according to their impact in predicting the response, or even their causal effect (Strobl et al., 2008). Our modeling approach allowed us to maintain all variables considered that could interact with bat occurrence. However, these values should not be understood as the estimated parameters as in the Linear Regression Models (LRMs), which act as weights in the predictions. For example, the interpretation of a weight in the linear regression model to the increase of a numerical feature by one unit changes the estimated outcome by its weight (in the case a numeric feature), which is not the interpretation in RUSBoost or others non-parametric methods. Also, variable importance scores only allows positive values, which does not occur in LRMs methods. The next section presents how to analyze the effect of a variable with respect to the response predicted by the model.

2.3.2 Effect of a variable on presence probability

We used Partial Dependence Plots (PDPs) (Elith et al., 2008; Friedman, 2001) to analyze relationships between explanatory variables and bat occurrence. PDP shows, in particular, the partial contribution of a given variable considering all other variables at their measured values, presented visually by line plot (2D graphic); or via a 3D surface. For example, if we separate the attribute vector $\vec{x}$ in a subset of attributes $\vec{x}_S$ and $\vec{x}_C$, where the latter vector represents the complement over $\vec{x}$, we get $\vec{x} = (\vec{x}_S, \vec{x}_C)$ and the expectation of predicted answers regarding $\vec{x}_C$ is naturally given by:

$$\mathcal{M}_{\vec{x}_S}(\vec{x}_S) = \mathbb{E}_{\vec{x}_C} [\mathcal{M}(\vec{x}_S, \vec{x}_C)] = \int \mathcal{M}(\vec{x}_S, \vec{x}_C) d\mathbb{P}(\vec{x}_C) \approx \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \mathcal{M}(\vec{x}_S, \vec{x}_C^i) \quad (8)$$

where $\vec{x}_C^i = (\vec{x}_S^i, \vec{x}_C^i)$ is the i -th observation. The last sum in Eq. (8) represents an approximation, assuming that each observation is equally likely. The PDP is defined as the graph of the function $\mathcal{M}_{\vec{x}_S}$, which we will denote by $\mathcal{G}[\mathcal{M}_{\vec{x}_S}]$. A PDP can show whether the relationship between the bat occurrence and a specific set of variables is linear, monotonic or more complex, and it can provide an important interpretability of how the variable to be evaluated is influencing (in our study) the probability of the species presence.

The summation in the Eq. (8) assumes the existence of the data points $(\vec{x}_S, \vec{x}_C^i)$, and then when the variables are correlated, such artificial samples may have a very low probability measure, therefore, before building our models, in the database of our case study, we eliminate highly correlated variables (> 0.5).

3 Results

3.1 Hyper-parameter tuning and modeling

We built models for each bat species ($|\mathcal{B}| = 25$) and obtained the values of the hyperparameters after the optimization process (see Table S1 of Supplementary Material for more details). After cross validation procedure, the averages values for AUC, specificity, sensitivity, and overall accuracy were 0.8 ± 0.1 , $79.6 \pm 4.9\%$, $78.8 \pm 5.0\%$, and $78.8 \pm 5.0\%$, respectively (see Fig. 3 and Table S1), and therefore, models predicted species presence better than random (AUC values > 0.5). The average values obtained for *MSN*, *NL* and *LR* were 2402.7 ± 2451.7 , 446.24 ± 102.92 and 0.1 ± 0.2 , respectively (see Table S1).

We analyzed the effect of the hyperparameters optimization procedure using the bat species with most and least presence. Therefore, we used the distribution of *P. pipistrellus* with a total presence of 37.2% (2297 records) and *P. nathusii* with a presence of 0.5% (33 records) (Fig. 5). In this particular case, the bayesian method iteratively explored the values until converging to a solution that minimized the the *kFoldLoss* described in the Eq. (5). Fig. 4 shows the best hyper-parameter values obtained when solving the optimization problem defined in the Eq. (7) for the species *P. pipistrellus*: $(z_{NMS}^*, z_{NL}^*, z_{LR}^*) = (108, 338, 0.96)$; and *P. nathusii*: $(z_{NMS}^*, z_{NL}^*, z_{LR}^*) = (89, 499, 0.01)$. These values were reached after iteration 23 for the model corresponding to Ppip (Fig. 4 - left side) and 13 for Pnat species (Fig. 4 - right side).

The minimum classification error (MCE) decreases with the number of iterations, until it stabilizes at an “optimal” value (Fig. 4).

Once the hyperparameters $\vec{z}^* = (z_{NMS}^*, z_{NL}^*, z_{LR}^*)$ have been established for each model/species, the $\mathcal{M}^{Ppip}$ and $\mathcal{M}^{Pnat}$ models are built and then used to calculate the presence probability in each cell of our study area (a total of 6,169 cells within Iberian Peninsula); Fig. 5 shows the result of this procedure. The maps located in the upper show the presence records of each species and the maps located in the lower the probabilities predicted by each model. Visually, we can see that presences and high values are properly correlated. In our Supplementary Material we present the probability maps for the rest.

Figure 3: Summary of performance models, where the left and right chart are expressed in percentage and the center in a range from 0 to 1.

Figure 4: Minimum classification error (MCE) evolution for Ppip (left chart) and Pnat (right chart). The MCE is updated as the optimization algorithm runs. Each light blue dot represents an estimate of the MCE value for a combination of hyper-parameter values $\vec{z} = (z_{MNS}, z_{NL}, z_{LR})$ tested until the current iteration. Similarly, dark blue dots indicate the observed minimum MCE. The red square and orange dot (the two marks coincide on the left graphic) indicate the best hyperparameters combination and those that produce the minimum observed MCE, respectively.

276 We observed performance metrics from Table S1, that after cross validation procedure, the averages
 277 values for AUC, specificity, sensitivity, and overall accuracy were 0.93, 89.7%, 78.8%, and 89.6%, for
 278 Pnat species respectively; while that for Ppip were 0.86, 84.0%, 71.0%, and 79.1%, respectively. In
 279 both cases, good values were obtained for all the metrics, however, the values were not proportional to
 280 the number of presence records. Broadly, we did not find a significant correlation between the number
 281 of species records with the performance measures of RUSBoost method.

Figure 5: RUSBoost estimations for Ppip and Pnat species

3.2 Importance of the variables

This section highlights the variable importance estimations in the construction of the models. In Fig. S1 of our Supplementary material the values obtained for each model/species are shown, of the thirty-six variables considered in our study, and for greater clarity, we summarized the results in the Fig. 6 and 7. Specifically, the first figure shows the variables corresponding to land cover (left chart) and weathers (right chart); and the second corresponds to topography and human activity. According to the figures, the land covers with the highest value of importance in the prediction of most of the models were the variables *Forest*, *Shurbland* and *Rainfallcrops*, whose medians exceed 3%. These variables are closely followed (below 3%) by *Pasture*, *Mixedcrops*, *Abandoned*, *Irrigatedcrops* and *Urban* covers. The remaining variables such as *Mines*, *Roads*, *River*, *Wetland*, *SandRock*, *Agroforestry* and *Improductive* are below the 2% importance value according to our results.

Regarding to weather drivers, most are in the range of 3% to 5%, except for the variables *pmr*, *pm24* and *perm* that represent the relative maximum precipitation; 24-hour maximum rainfall; and soil permeability respectively. Only the variable *inso* exceeds 4% of importance (its median), and then the variables of relative humidity in January (*huen*), average annual evapotranspiration (*etr*), average annual temperature (*tmed*), average July temperature (*tjul*), and so forth.

Finally, in Fig. 7 presents the values obtained for topography (left chart) and human activity (right chart) drivers. All topographic variables are in the range of 3 to 5 percent, although their medians are concentrated below 4%. On the other hand, in the human activity variables it can be appreciated that only the *daut* variable exceeds 3% while the other variables (*u100* and *u500*) are below this value.

4 Discussion

4.1 Handling class imbalance

The ability of handling class imbalance is a necessary skill in our study, since the variability of record numbers for each bat on each trained the model is small, due to the low sighting frequency of each species (see Fig. 1 and Table S1). In our study, we have applied RUSBoost to a data of

Figure 6: Importance of variables of Land cover and Weather drivers.

Figure 7: Importance of variables of Topography and Human Activity drivers.

308 presence/absence of bats in a vast region and with very uneven and imbalanced records. However,
 309 the method, independently of the number of records, obtained an excellent performance (see Fig. 8),
 310 comparable to other similar studies (see e.g. Maher et al. (2014)).

311 Applying methods such as Random Forest or AdaBoost are not suitable when classes in categorical
 312 response variables are imbalanced (Chen, 2004) due to bootstrap over-representing the majority class,
 313 leading to under-prediction of the minority class, so the overall performance measures (AUC and
 314 Overall Accuracy) can be misleading. Currently, there are three ways to deal with imbalanced data:
 315 i) assigning a high cost to minority class misclassification (Chen, 2004); ii) reducing the sample of the
 316 majority class (Kubat et al., 1998); and iii) oversampling the minority class (Chawla et al., 2002).
 317 RUSBoost method relies on the second approach balancing the response variable that iteratively
 318 down-samples the majority class by randomly drawing $2 \cdot R$ (where R is the size of minority class, see
 319 Fig. 2) from the majority class and building decision tree models iteratively using different random
 320 subsets while holding the sample-size of the minority-class constant (Evans and Cushman, 2009).

321 The RUSBoost method has been successfully applied in different contexts: to determine impervi-
 322 ous surface areas (Kesikoglu et al., 2016); to identify dolphins species from characteristics extracted
 323 from photos (Maglietta et al., 2018); and was recently used to detect electrical losses in power systems
 324 Adil et al. (2020), which is compared with state-of-the-art methods showing the superiority for han-
 325 dling imbalanced data, parameter optimization and overfitting. Also, recently RUSBoost was used to
 326 assess risks of fire (or fire occurrence probability) for territorial planning (Carrasco et al., 2021), but

Figure 8: Number of presence records vs performance metrics.

the dataset was slightly imbalanced and contained only a type of “species” (fire and non-fire). However, to our knowledge in few references we have been able to find its use in species distribution models in a few references. A review of dedicated classification methods when the database is imbalanced can be found in [Haixiang et al. \(2017\)](#).

4.2 Hyper-parameter optimization

Every machine learning algorithm or software has hyper-parameters, and a desirable task is to automate them in order to optimize model performance. However, this is not an easy task, and the complexity of this process will mainly depend on the nature of the variables/hyper-parameters and the objective function to be optimized (e.g. performance measure), where the latter is usually a black-box function. Perhaps the most common algorithms to perform this task are: Grid Search ([Bergstra et al., 2011](#)), Random Search ([Bergstra and Bengio, 2012](#)), Genetic Algorithm ([Lessmann et al., 2005](#)) and Bayesian Optimization method (BOM) ([Eggensperger et al., 2013](#)). Contrasting with traditional optimization methods such as gradient descent based, for example: NEWUOA ([Powell, 2006](#)), BOBYQA ([Powell, 2009](#)), DFO ([Conn and Toint, 1996](#)), and so forth; Many of these algorithms also have the capacity to effectively identify discrete, categorical, and conditional hyper-parameters.

In algorithmic and very general terms, Grid Search (GS) is based on the concept of defining a fixed hyper-parameter search space and then detecting the hyper-parameter combinations in it, in order to find the best performing hyper-parameter combination. On the other hand, Random search (RS) randomly selects combinations of hyper-parameters in the search space (which may not be fixed), given limited execution time and resources. In GS and RS, each hyper-parameter configuration is treated independently and therefore can easily be parallelized. Unlike GS and RS, BOM determines the next hyper-parameter value based on the previous results of tested hyper-parameter values, which avoids many unnecessary evaluations; thus, BOM can detect the optimal hyper-parameter combination within fewer iterations rather than GS and RS ([Yang and Shami, 2020](#)). A comprehensive review of algorithms and applications can be found in [Yu and Zhu \(2020\)](#).

Broadly, there is no criterion to evaluate if the solution (combination of hyper-parameters found) is the optimal solution of the optimization problem defined in the Eq. (7) because the $kFoldLoss$ function (see Eq. (5)) is a black-box function and therefore we do not know about its structural mathematical

properties (continuity, differentiability, etc). On the other hand, the number of iterations is predefined by the user, in our study in particular, a maximum number of 30 iterations were arbitrarily established, due to the computational time consumption (approximately one minute per iteration).

4.2.1 Variable interaction on presence probability

In this section, we discuss the potential of our methodology to analyze the probability sensitivity of bat occurrence, with respect to the variation of a variable through an illustrative example (see Section 2.3.2). For them, we built the functions $\mathcal{M}_{\vec{x}_S}(\vec{x}_S) := \mathbb{E}_{\vec{x}_C} [\mathcal{T}(\vec{x}_S, \vec{x}_C)]$ for S representing individual variables, i.e. $|S| = 1$ (see Eq. (8)). Strictly speaking, we could have carried out this analysis for each of the variables and for each of the species considered in our study, producing a total of $36 \times 25 = 900$ PDPs. However, for practical and expository purposes, we only include the analysis of the effect of the variable *tjul* on the probability of the presence of the species *Barbastella barbastellus* (Bbar).

Because RUSBoost algorithm involves Random Undersampling of data for building the ensemble model detailed rule sets can vary slightly between different model runs, and therefore it is advisable to run multiple bootstrap iterations of algorithm to achieve stable and repeatable results (Murphy et al., 2010). The influence of the temperature variable on the predicted presence probability is visualized in Fig. 9. The plot shows a slight decrease in probability from 14 to 21 Celsius' degrees, and a steep decrease from 21 to 26 degrees. Above 26 degrees the probability stabilizes. The figure also shows that the probability values remain within an acceptable error (see standard deviation in the chart), under different replications for the construction of the model, and therefore the mean values (blue-line) can be taken as a reliable estimate.

Figure 9: Partial dependence plot for *tjul* variable and *Bbar* species.

4.2.2 Bayesian interpretation

From Eq. (1), weighted voting schemes can be seen as approximations under a Bayesian approach. In fact, let $\mathcal{Y}$ be the random variable which represents the choice of the presence/absence class, and informally we denote $\mathbb{P}(\mathcal{M}_l) = \alpha_l$ as the discrete probability distribution selection of decision trees in the RUSBoost procedure. Replacing $\mathcal{M}_l(\vec{x})$ in the Eq. (1) with the class posterior probabilities denoted by $\mathbb{P}(\mathcal{Y} | \vec{x}, \mathcal{M}_l)$ (or relative frequency of classes, before the tree leaf in the decision tree model $\mathcal{M}_l$). From this, the Eq. (1) (right side) can be rewritten as follows:

$$\mathbb{P}(\mathcal{Y} | \vec{x}) = \sum_{l=1}^L \mathbb{P}(\mathcal{Y} | \vec{x}, \mathcal{M}_l) \mathbb{P}(\mathcal{M}_l). \quad (9)$$

Eq. (9) helps understand the intuition behind the previously defined probability of bat occurrence (see Eq. (1)) as a probability that is conditional on the variables of a observed point $\vec{x}$, and it is obtained as a weighted summation of the class posterior and decision trees selection probabilities.

5 Conclusions

The RUSBoost method assessed in this study yielded similar performance values (AUC, sensitivity, specificity and overall accuracy) to common techniques and, thus, expand modelling options for ecologists. We found that the performance of the method is invariant to the number of presence records (see Fig. 8) and therefore it can be applied as a method based on presence-only data, without losing the underlying abilities of the presence/absence methods, for example, enabling analyses of biases and prevalence (Phillips et al., 2009). It is important to emphasize that the method is suitable when the absence data is reliable. When this is not the case or the species not easily detected, it is preferable to use presence-only techniques such as MaxEnt (Phillips et al., 2006), GARP (Genetic Algorithm for Rule-set Production) (Stockwell, 1999) or SVMs (Support Vector Machines) (Guo et al., 2005). The method showed broadly a good balance between sensitivity and specificity, a very important aspect for rare and threatened species and conservation decisions (Guisan et al., 2013).

We also show that the method is sensitive to its hyper-parameters. However, they can be optimized with the Bayesian Optimization Method in a reasonable time. In the case of the model built for the Pnat species, the overall precision of the method increased from 83% to 90% in less than 15 iterations (approximately 15 minutes on our computer), which shows the advantages of combining the RUSBoost with hyper-parameter optimization.

Authors' contributions

J.C., F.L. and A.W. conceived the project; F.L. collected the data; J.C. and F.L. raised the scientific hypotheses and methodology; J.C. wrote the code; all authors contributed to writing the manuscript, with J.C. having the lead on the writing. F.L. and A.W. reviewed and edited the final manuscript.

Acknowledgements

We thank Ángeles Haz for check and correct the English version of the manuscript. Jaime Carrasco acknowledges the support of the Agencia Nacional de Investigación y Desarrollo (ANID), Chile, through funding Postdoctoral Fondecyt project 3210311. Fulgencio Lisón was supported by Fondecyt iniciacion project N° 11180514, Andrés Weintraub thanks Fondecyt 1170381.

References

- M. Adil, N. Javaid, U. Qasim, I. Ullah, M. Shafiq, and J.-G. Choi. Lstm and bat-based rusboost approach for electricity theft detection. *Applied Sciences*, 10(12):4378, 2020.
- A. M. Barbosa, M. A. Puerto, D. Figueiredo, and R. Real. Modelling disjunct carnivore distributions: the case of the wolf (*canis lupus*) in the iberian peninsula. *Carnivores: species, conservation and management (FI Alvares and GE Mata, eds.)*. Nova Publishers, New York, pages 119–137, 2012.
- J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.
- J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.

- 421 C. M. Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.
- 422 L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.
- 423 L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- 424 L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and regression trees*.
425 Routledge, 2017.
- 426 L. Breiman et al. Statistical modeling: The two cultures (with comments and a rejoinder by the
427 author). *Statistical science*, 16(3):199–231, 2001.
- 428 F. T. Breiner, A. Guisan, A. Bergamini, and M. P. Nobis. Overcoming limitations of modelling rare
429 species by using ensembles of small models. *Methods in Ecology and Evolution*, 6(10):1210–1218,
430 2015.
- 431 L. Brotons, W. Thuiller, M. B. Araújo, and A. H. Hirzel. Presence-absence versus presence-only
432 modelling methods for predicting bird habitat suitability. *Ecography*, 27(4):437–448, 2004.
- 433 J. Busby. Bioclim-a bioclimate analysis and prediction system. *Plant protection quarterly*, 61:8–9,
434 1991.
- 435 J. Carrasco, M. Acuna, A. Miranda, G. Alfaro, C. Pais, and A. Weintraub. Exploring the multidimen-
436 sional effects of human activity and land cover on fire occurrence for territorial planning. *Journal*
437 *of Environmental Management*, 297:113428, 2021.
- 438 D. Castelvocchi. Can we open the black box of ai? *Nature News*, 538(7623):20, 2016.
- 439 N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-
440 sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.
- 441 C. Chen. Using random forest to learn imbalanced data. 2004.
- 442 W. W. Cohen. Fast effective rule induction. In *ICML*, 1995.
- 443 A. R. Conn and P. L. Toint. An algorithm using quadratic interpolation for unconstrained derivative
444 free optimization. In *Nonlinear optimization and applications*, pages 27–47. Springer, 1996.
- 445 W. K. Cornwell, D. W. Schilck, and D. D. Ackerly. A trait-based test for habitat filtering: convex
446 hull volume. *Ecology*, 87(6):1465–1471, 2006.
- 447 D. R. Cutler, T. C. Edwards Jr, K. H. Beard, A. Cutler, K. T. Hess, J. Gibson, and J. J. Lawler.
448 Random forests for classification in ecology. *Ecology*, 88(11):2783–2792, 2007.
- 449 G. De’Ath. Boosted trees for ecological modeling and prediction. *Ecology*, 88(1):243–251, 2007.
- 450 J. M. Drake and J. M. Bossenbroek. Profiling ecosystem vulnerability to invasion by zebra mussels
451 with support vector machines. *Theoretical Ecology*, 2(4):189–198, 2009.
- 452 J. M. Drake, C. Randin, and A. Guisan. Modelling ecological niches with support vector machines.
453 *Journal of applied ecology*, 43(3):424–432, 2006.
- 454 K. Eggenberger, M. Feurer, F. Hutter, J. Bergstra, J. Snoek, H. Hoos, K. Leyton-Brown, et al.
455 Towards an empirical foundation for assessing bayesian optimization of hyperparameters. In *NIPS*
456 *workshop on Bayesian Optimization in Theory and Practice*, volume 10, 2013.

- 457 J. Elith and J. R. Leathwick. Species distribution models: ecological explanation and prediction across
458 space and time. *Annual review of ecology, evolution, and systematics*, 40:677–697, 2009.
- 459 J. Elith, C. H. Graham*, R. P. Anderson, M. Dudík, S. Ferrier, A. Guisan, R. J. Hijmans,
460 F. Huettmann, J. R. Leathwick, A. Lehmann, et al. Novel methods improve prediction of species’
461 distributions from occurrence data. *Ecography*, 29(2):129–151, 2006.
- 462 J. Elith, J. R. Leathwick, and T. Hastie. A working guide to boosted regression trees. *Journal of*
463 *Animal Ecology*, 77(4):802–813, 2008.
- 464 J. S. Evans and S. A. Cushman. Gradient modeling of conifer species using random forests. *Landscape*
465 *ecology*, 24(5):673–683, 2009.
- 466 P. A. Flach. On the state of the art in machine learning: A personal review. *Artificial Intelligence*,
467 131(1-2):199–222, 2001.
- 468 Y. Freund, R. E. Schapire, et al. Experiments with a new boosting algorithm. In *icml*, volume 96,
469 pages 148–156. Citeseer, 1996.
- 470 J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*,
471 pages 1189–1232, 2001.
- 472 D. García-Callejas and M. B. Araújo. The effects of model and data complexity on predictions from
473 species distributions models. *Ecological Modelling*, 326:4–12, 2016.
- 474 I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.
- 475 A. Guisan, R. Tingley, J. B. Baumgartner, I. Naujokaitis-Lewis, P. R. Sutcliffe, A. I. Tulloch, T. J. Re-
476 gan, L. Brotons, E. McDonald-Madden, C. Mantyka-Pringle, et al. Predicting species distributions
477 for conservation decisions. *Ecology letters*, 16(12):1424–1435, 2013.
- 478 Q. Guo, M. Kelly, and C. H. Graham. Support vector machines for predicting distribution of sudden
479 oak death in california. *Ecological modelling*, 182(1):75–90, 2005.
- 480 G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing. Learning from class-
481 imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73:
482 220–239, 2017.
- 483 A. Jiménez-Valverde, P. Acevedo, A. M. Barbosa, J. M. Lobo, and R. Real. Discrimination capacity in
484 species distribution models depends on the representativeness of the environmental domain. *Global*
485 *Ecology and Biogeography*, 22(4):508–516, 2013.
- 486 R. A. Johnson, N. V. Chawla, and J. J. Hellmann. Species distribution modeling and prediction: A
487 class imbalance problem. In *2012 Conference on Intelligent Data Understanding*, pages 9–16. IEEE,
488 2012.
- 489 J. Juste, M. Ruedi, S. J. Puechmaille, I. Salicini, and C. Ibáñez. Two new cryptic bat species within
490 the myotis nattereri species complex (vespertilionidae, chiroptera) from the western palaeartic.
491 *Acta Chiropterologica*, 20(2):285–300, 2018.
- 492 M. Kesikoglu, U. Atasever, C. Ozkan, and E. Besdok. The usage of rusboost boosting method for
493 classification of impervious surfaces. *International Archives of the Photogrammetry, Remote Sensing*
494 *& Spatial Information Sciences*, 40, 2016.
- 495 M. Kubat, R. C. Holte, and S. Matwin. Machine learning for the detection of oil spills in satellite
496 radar images. *Machine learning*, 30(2):195–215, 1998.

- 497 S. Lek and J.-F. Guégan. *Artificial neuronal networks: application to ecology and evolution*. Springer
498 Science & Business Media, 2012.
- 499 S. Lessmann, R. Stahlbock, and S. F. Crone. Optimizing hyperparameters of support vector machines
500 by genetic algorithms. In *IC-AI*, pages 74–82, 2005.
- 501 F. Lisón and D. Sánchez-Fernández. Low effectiveness of the natura 2000 network in preventing
502 land-use change in bat hotspots. *Biodiversity and Conservation*, 26(8):1989–2006, 2017.
- 503 F. Lisón, D. Sánchez-Fernández, and J. F. Calvo. Are species listed in the annex ii of the habitats
504 directive better represented in natura 2000 network than the remaining species? a test using spanish
505 bats. *Biodiversity and Conservation*, 24(10):2459–2473, 2015.
- 506 R. Maglietta, A. Bruno, V. Renò, G. Dimauro, E. Stella, C. Fanizza, S. Bellomo, G. Cipriano, A. Tursi,
507 and R. Carlucci. The promise of machine learning in the risso’s dolphin grampus griseus photo-
508 identification. In *2018 IEEE International Workshop on Metrology for the Sea; Learning to Measure*
509 *Sea Health Parameters (MetroSea)*, pages 183–187. IEEE, 2018.
- 510 S. P. Maher, C. F. Randin, A. Guisan, and J. M. Drake. Pattern-recognition ecological niche models
511 fit to presence-only and presence-absence data. *Methods in Ecology and Evolution*, 5(8):761–770,
512 2014.
- 513 D. Mladenic and M. Grobelnik. Feature selection for unbalanced class distribution and naive bayes.
514 In *ICML*, 1999.
- 515 M. A. Murphy, J. S. Evans, and A. Storfer. Quantifying bufo boreas connectivity in yellowstone
516 national park with landscape genetics. *Ecology*, 91(1):252–261, 2010.
- 517 J. D. Olden, J. J. Lawler, and N. L. Poff. Machine learning methods without tears: a primer for
518 ecologists. *The Quarterly review of biology*, 83(2):171–193, 2008.
- 519 L. J. Palomo, J. Gisbert, and J. C. Blanco. *Atlas y libro rojo de los mamíferos terrestres de España*.
520 Organismo Autónomo de Parques Nacionales Madrid, Spain, 2007.
- 521 S. J. Phillips, R. P. Anderson, and R. E. Schapire. Maximum entropy modeling of species geographic
522 distributions. *Ecological modelling*, 190(3-4):231–259, 2006.
- 523 S. J. Phillips, M. Dudík, J. Elith, C. Graham, A. Lehmann, J. Leathwick, and S. Ferrier. Sample
524 selection bias and presence-only distribution models: implications for background and pseudo-
525 absence data. *Ecological applications : a publication of the Ecological Society of America*, 19 1:
526 181–97, 2009.
- 527 M. J. Powell. The newuoa software for unconstrained optimization without derivatives. In *Large-scale*
528 *nonlinear optimization*, pages 255–297. Springer, 2006.
- 529 M. J. Powell. The bobyqa algorithm for bound constrained optimization without derivatives. *Cam-*
530 *bridge NA Report NA2009/06, University of Cambridge, Cambridge*, pages 26–46, 2009.
- 531 H. Pulliam. On the relationship between niche and distribution. *Ecology Letters*, 3:349–361, 2000.
- 532 J. R. Quinlan. C4.5: Programs for machine learning. 1992.
- 533 A. Rainho, P. Alves, F. Amorim, and J. T. Marques. *Atlas dos morcegos: de Portugal continental*.
534 ISBN 978-972-775-226-3. OCLC: 960423369.
- 535 B. Richhariya and M. Tanveer. A reduced universum twin support vector machine for class imbalance
536 learning. *Pattern Recognition*, 102:107150, 2020.

537 O. J. Robinson, V. Ruiz-Gutierrez, and D. Fink. Correcting for bias in distribution modelling for rare
538 species using citizen science data. *Diversity and Distributions*, 24(4):460–472, 2018.

539 B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support
540 of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

541 C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano. Rusboost: A hybrid approach to
542 alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems
543 and Humans*, 40(1):185–197, 2009.

544 N. Sillero, S. Arenas-Castro, U. Enriquez-Urzelai, C. G. Vale, D. Sousa-Guedes, F. Martínez-Freiría,
545 R. Real, and A. M. Barbosa. Want to model a species niche? a step-by-step guideline on correlative
546 ecological niche modelling. *Ecological Modelling*, 456:109671, 2021.

547 J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning
548 algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

549 J. Soberón and M. Nakamura. Niches and distributional areas: Concepts, methods, and assumptions.
550 *Proceedings of the National Academy of Sciences*, 106:19644 – 19650, 2009.

551 M. Soley-Guardia, E. E. Gutiérrez, D. M. Thomas, J. Ochoa-G, M. Aguilera, and R. P. Anderson.
552 Are we overestimating the niche? removing marginal localities helps ecological niche models detect
553 environmental barriers. *Ecology and evolution*, 6(5):1267–1279, 2016.

554 D. Stockwell. The garp modelling system: problems and solutions to automated spatial prediction.
555 *International journal of geographical information science*, 13(2):143–158, 1999.

556 C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis. Conditional variable importance
557 for random forests. *BMC bioinformatics*, 9(1):307, 2008.

558 E. Thibaud, B. Petitpierre, O. Broennimann, A. C. Davison, and A. Guisan. Measuring the relative
559 effect of factors affecting species distribution model predictions. *Methods in Ecology and Evolution*,
560 5(9):947–955, 2014.

561 A. Tsoar, O. Allouche, O. Steinitz, D. Rotem, and R. Kadmon. A comparative evaluation of presence-
562 only methods for modelling species distribution. *Diversity and distributions*, 13(4):397–405, 2007.

563 J. A. Velasco and C. González-Salazar. Akaike information criterion should not be a “test” of geo-
564 graphical prediction accuracy in ecological niche modelling. *Ecological Informatics*, 51:25–32, 2019.

565 M. S. Wisz, J. Pottier, W. D. Kissling, L. Pellissier, J. Lenoir, C. F. Damgaard, C. F. Dormann,
566 M. C. Forchhammer, J.-A. Grytnes, A. Guisan, et al. The role of biotic interactions in shaping
567 distributions and realised assemblages of species: implications for species distribution modelling.
568 *Biological reviews*, 88(1):15–30, 2013.

569 L. Yang and A. Shami. On hyperparameter optimization of machine learning algorithms: Theory and
570 practice. *Neurocomputing*, 415:295–316, 2020.

571 T. Yu and H. Zhu. Hyper-parameter optimization: A review of algorithms and applications. *arXiv
572 preprint arXiv:2003.05689*, 2020.