Expert Systems With Applications 202 (2022) 117107

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Regression random machines: An ensemble support vector regression model with free kernel choice

Anderson Ara [a,*], Mateus Maia [b], Francisco Louzada [c], Samuel Macêdo [d]

[a] Department of Statistics, Federal University of Paraná, Curitiba, Paraná, Brazil
[b] Department of Math & Statistics, Maynooth University, Maynooth, Ireland
[c] Institute of Mathematical and Computer Sciences, University of São Paulo, São Carlos, São Paulo, Brazil
[d] Department of Natural Sciences and Mathematics, Federal Institute of Pernambuco, Recife, Pernambuco, Brazil

ARTICLE INFO

Keywords:
Support vector regression
Bagging
Kernel functions

ABSTRACT

Machine learning techniques have one of their main objectives to reduce the generalized prediction error. Support vector models have as a main challenge the choice of an appropriate kernel function, as well as the estimation of its hyperparameters. Such procedures are usually performed through some tests and tuning processes which require a high computational performance. In contrast, ensemble methods present a good approach to combine several models which result in a greater predictive capacity. In this paper, we propose a new ensemble method to support vector regression, namely regression random machines. The proposed method eliminates the need to choose the best kernel function during the tuning process using a random mixture of kernel functions combined with a properly bagging ensemble which considers the strength and agreement of the single models. The results demonstrated a good predictive performance through lower generalization error which overlaps the single and bagged versions of support vector models with different kernels. The usefulness of the proposed method is illustrated by simulation studies that were realized over eight artificial scenarios and twenty-seven real-world applications.

1. Introduction

The prediction of new observations or events through statistical models is one of the main objectives of supervised statistical learning methods. Currently, machine learning models have several applications in regression tasks in a wide range of science fields, for instance, economy — predicting bitcoin's price (McNally, Roche, & Caton, 2018), biology — predicting biological properties from plants (Féret et al., 2019) or classifying gene functions (Park, Koo, Kim, Sohn, & Lee, 2008), and physics — predicting electrical propertiesfrom materials (Chen, Tran, Batra, Kim, & Ramprasad, 2019). Inside this type of regression models, there is the support vector regression (SVR) model that was proposed by Drucker, Burges, Kaufman, Smola, and Vapnik (1997) and has been used extensively as an optimal solution when compared with other traditional base-line methods (Delbari, Sharifazari, & Mohammadi, 2019; Khosravi, Koury, Machado, & Pabon, 2018; Wu, Ho, & Lee, 2004; Xiao, Zhang, Zhong, Shao, & Li, 2018). Moreover, the SVR has good properties that differ from other models such as convex optimization and the foundation of statistical learning theory. (Shivaswamy, Chu, & Jansche, 2007) attribute the success of support vector models mainly to four factors: (i) rooted in the statistical learning theory, SVMs possess superior generalization capacity; (ii) a globally optimal solution is obtainable by solving a convex optimization problem, while the problems of local minima impede other contemporary approaches, such as neural networks; (iii) using the so-called kernel trick, it is convenient to solve non-linear problems in arbitrarily high dimensional feature spaces; (iv) only a part of training samples are involved in solution representation.

Ensemble learning is also a predictive modeling strategy that combines models in order to achieve greater predictive capacity. The combination of singular models can enhance predictive power and increase its generalization power (Van Wezel & Potharst, 2007). Even novel approaches, as deep learning models, can also benefit from ensemble procedures (Araque, Corcuera-Platas, Sánchez-Rada, & Iglesias, 2017). In general, there are two main types of ensemble algorithms: bagging (Breiman, 1996) that uses independent bootstrap samples to create multiple models and built a final classifier by average mean or majority vote, reducing the variance, and boosting algorithms (Freund, Schapire, & Abe, 1999) that built sequential models directed to assign different weights based on their errors. There are several works that

* Corresponding author.
E-mail addresses: ara@ufpr.br (A. Ara), mateus.maiamarques.2021@mumail.ie (M. Maia), louzada@icmc.usp.br (F. Louzada), samuelmacedo@recife.ifpe.edu.br (S. Macêdo).

https://doi.org/10.1016/j.eswa.2022.117107
Received 12 August 2021; Received in revised form 23 February 2022; Accepted 28 March 2022
Available online 9 April 2022

present frameworks to improve the ensemble approaches selecting the models which are the most accurate and diverse (Khan et al., 2019) or adapting them to specific data scenarios, such as high dimensional data (Brahim & Limam, 2018).

The bagging method does not require a specific type of base classifier and can be used to improve predictions in regression tasks (Borra & Di Ciaccio, 2002; Mendes-Moreira, Soares, Jorge, & Sousa, 2012; Rakesh & Suganthan, 2017). This method can be used to enhance a single support vector regression model and other kinds of algorithms. The bagging approach using support vector regression models is already reported in literature through diverse applications. For example, to predict protein retention time (Song et al., 2002), to predict time series (Deng, Jin, & Zhong, 2005), electric load forecasting (Li, Che, & Yang, 2018), to forecast building occupation (Wu et al., 2018) and to predict blood pressure measures (Lee, Ahmad, & Jeon, 2018).

Despite the different number of works that present the bagging based on support vector regression models, there is no proposal of standard framework to choose which kernel function will be used in ensembles that use SVR as base-learners. In support vector models the kernel function and its hyperparameters have a compelling impact on the efficiency of the algorithm (Jebara, 2004). Generally, this selection is made by a grid-search, which choose those parameters that produces the lower test error inside a grid of possible combinations, by random search (Bergstra & Bengio, 2012), a stochastic combination of kernel parameters or functions (Mazaheri et al., 2019), or by Bayesian optimization algorithms (Bergstra, Bardenet, Bengio, & Kégl, 2011). All of them are computationally expensive and can consume a high computational time.

Moreover, there is an extensive discussion about the strength and diversity trade-off in ensemble learning methods (Bhatnagar, Bhardwaj, Sharma, & Haroon, 2014; Bi, 2012; Cunningham & Carney, 2000; Kuncheva, 2003; Lam, 2000; Sluban & Lavrač, 2015). Essentially, to achieve a better result with the combination of multiple models is necessary to have great strength and a high diversity simultaneously among the base learners. The strength of a model can be defined as its predictive capacity, and the diversity is related to a degree of independence between models, in other words, different classifiers complement one another as sources of evidence of the correct classification of random objects. The setting to optimize an ensemble approach regarding these two features is still an open issue with different approaches to handle it (Alzubi, 2015; Brown, Yao, Wyatt, Wersing, & Sendhoff, 2002; Cavalcanti, Oliveira, Moura, & Carvalho, 2016; Chandra & Yao, 2006; Smutz & Stavrou, 2016; Travis-Lumer & Goldberg, 2021).

In classification contexts, ensemble and hybrid methods with support vector machines using different kernels functions has been discussed in recent literature. Evgeniou, Pontil, and Elisseeff (2004) analyze the combinations of the support vector machines, as a special case of kernel machines model, and present theoretical estimates of their leave-one-out error. Wang, Zheng, Yoon, and Ko (2018) propose a SVM ensemble with bagging that each base learner uses a 2-d polynomial kernel function. (Wang et al., 2018) proposed a hybrid algorithm using twelve SVM base learners. Mazaheri et al. (2019) propose an algorithm relies on a voting procedure among stochastically generated kernel classifiers. Ara, Maia, Louzada, and Macêdo (2021a) propose the Random Machines method, a novel framework to deal with the kernel function selection in classification tasks.

This paper introduces a novel machine learning method that presents a solution for the choice of kernel function to be used in the bagged supported vector regression, using an alternative to the open problem of kernel and hyperparameters' selections. This novel method, namely Regression Random Machines (RRM), gives a solution for the kernel function's choice and tuning processes with efficient computational time and robust predictive power, and that improves the diversity of the combination without worsening its prediction performance. The method received this name because it uses weighted random kernel choice for each model that composes the aggregation of support vector regression learners, increasing the predictive power of the final model. In general, the contributions of this paper are four-fold. (1) Propose RRM as a strong predictive model in regression machine learning method. (2) Eliminate the kernel turning process in support vector machine models. (3) Clarify the trade-off between strength and diversity to improve the predictive performance in ensemble models. (4) Traditional support vector regression and bagged support vector regression are particular cases of the random machines.

Table 1
Kernel functions.

Kernel	K(x,y)	Parameters
Linear Kernel	$\gamma(x \cdot y)$	γ
Polynomial Kernel	$(\gamma(x \cdot y))^d$	γ, d
Gaussian Kernel	$e^{-\gamma\|x-y\|^2}$	γ
Laplacian Kernel	$e^{-\gamma\|x-y\|}$	γ

The results were validated over simulation studies and diverse benchmarking datasets. The proposed approach is different from the traditional ensemble of support vector regression (SVR) models because the randomness used increases the diversity of base learners without reducing its predictive power. The idea of increasing the diversity and maintaining the accuracy in bagging was also demonstrated in works that use kNN classifiers as base models (Gul et al., 2018).

The rest of the paper is organized as follows: Section 2 presents a theoretical description review about the support vector machine method, proposed by Drucker et al. (1997), the challenges on the selection of hyperparameters and standard kernel functions and the traditional ensemble approaches. Section 3 introduces the proposed regression random machines (RRM) approach and how and why it works in detail of strength and diversity. Section 4 displays the results and applications on artificial and real datasets. Finally, Section 5 closes the paper with final remarks.

2. Support vector regression

Support Vector Machine (SVM) models were firstly introduced by Boser, Guyon, and Vapnik (1992) and Cortes and Vapnik (1995) in a context of classification tasks and became strongly relevant in the statistical learning field since they can achieve lower generalization error and have strong theoretical properties. Therefore, it theoretically guaranteed that it achieves the global minimum, while other algorithms as Neural Networks (ANN) can be trapped in local minima. Supposing a context in which the target variable y is continuous, support vector regression machines (SVR) (Drucker et al., 1997) have been proposed as a generalized version of support vector machines to regression tasks. Support vector models, classification and regression tasks, use the kernel trick to handle non-linear scenarios.

The functions $K(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{y})$ are defined as the semi-definite kernel functions (Courant & Hilbert, 1953) and various types of kernel functions can be used in distinct regression examples. The choice of particular kernel functions provides unique nonlinear mappings and the performance of the resulting SVR often depends on the appropriate choice of the kernel (Jebara, 2004). There are several kernel functions in the general framework for SVR, which some of the most common were used in this paper. They are presented in Table 1, and have hyperparameters γ and d, which $\gamma > 0$ and $d \in \mathbb{N}$.

Determining which the best kernel by grid search, or other search methods, can be an expensive and harrowing computational problem (Chapelle & Vapnik, 2000). In order to solve it, many works have tried to develop a methodology that can improve the selection of the best kernel function (Ayat, Cheriet, & Suen, 2005; Cherkassky & Ma, 2004; Friedrichs & Igel, 2005; Jebara, 2004; Wu, Tzeng, & Lin, 2009). Regression random machines method proposes an efficient alternative to work through a framework where it is avoidable for this exhaustive search, saving computational power and time, since this selection is made through a random process.

2.1. Bagging support vector regression

Bagging, an acronym of Bootstrapping Aggregation, is a popular ensemble procedure proposed by Breiman (1996). In general, the bagging generates data sets by random sampling with replacement from the training set with the same size n, also known as bootstrap samples. Then, each model $h_j(x_i)$ is trained independently for each bootstrapping sample j, $\forall j \in \{1, \dots, B\}$, where $h_i(\mathbf{x})$ is the model generated to each bootstrap sample from $i = 1, \dots, B$, and B is the number of total bootstrap samples.

Considering the bagging procedure, any model can be used as a base learner and it can improve the predictive power of non-parametric regression methods (Borra & Di Ciaccio, 2002). One possibility is to use the SVR as the base model to lower the generalization error. The use of the bagged SVR for regression tasks can be listed: content-based image retrieval (Yildizer, Balci, Hassan, & Alhajj, 2012), solar power forecasting (Abuella & Chowdhury, 2017), quantifying urban land cover (Okujeni, van der Linden, Suess, & Hostert, 2016), wind power prediction (Heinermann & Kramer, 2014) and a trimmed bagging approach (Croux, Joossens, & Lemmens, 2007).

Although some works applied bagged SVR, none of them present a general framework to deal with the choice of the best kernel function. Often, this choice is made by trial evaluation, by grid search, or by random search. As this proceeding is computationally expensive (Chapelle & Vapnik, 2000). Moreover, the lack of diversity between support vector models implies a weak improvement from the traditional bagging procedure performance (Kim, Pang, Je, Kim, & Bang, 2002).

3. Regression random machines

The Random Machines (RM) methodology is not a traditional ensemble learning, this method can be interpreted as a mixing of bagging and boosting focused on support vector models (Ara et al., 2021a). This section describes the method in detail.

Given a training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ with $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$, $\forall i = 1, \dots, n$; the kernel bagging method initialize by training single models $h_r(\mathbf{x})$, where $r = 1, \dots, R$, and R is the total number of different kernel functions that could be used in support vector regression models. For example, if $R = 4$ a possible choice is to define h_1 as SVR with *Linear kernel*, h_2 as SVR with *Polynomial kernel*, h_3 as SVR with *Gaussian kernel* and h_4 as SVR with *Laplacian kernel*.

Each model is validated for the test set $\{(\boldsymbol{x}_k, y_k)\}_{k=1}^{T}$, and the root mean squared error ($RMSE_r$), which we will refer as δ_r, is calculated for each model, $\forall r = 1, \dots, R$, where R is the number of kernel functions that will be used. As the range of the dependent variable in regression (y) is broad, the vector of root means squares $\boldsymbol{\delta}$ is divided by its deviation in order to standardize the error. Afterwards, sample probabilities, λ_r, are calculated by Eq. (1) for each kernel function

$$\lambda_r = \frac{e^{-\beta\delta_r}}{\sum_{i=1}^{R} e^{-\beta\delta_i}}, \tag{1}$$

with $\forall r = 1, \dots, R$.

Subsequently, B bootstrap samples are sampled from the training set. A support vector regression model g_k is trained for each bootstrap sample, $k = i, \dots, B$ and the kernel function that will be used for g_k will be determined by a random choice with probability $\lambda_r, \forall r = 1, \dots, R$. The probabilities λ_r are higher if the determined kernel function used in $h_r(\mathbf{x})$ has a lower generalization error measured from the calculated RMSE over the test set. Consequently, the models with lower δ_r will frequently appear when the random kernel selection for each bootstrap model is done.

The parameter β, named as the correlation parameter, will tune the penalty of the generalization error of each model. Fig. 1 shows that small values of β create heavy-tail penalty functions while greater beta's values represent light-tail penalty. The parameter gets its name because it can determine the diversity between the chosen kernel functions since high values further penalize the performance differences between each SVR model type. For instance, considering a value of $\beta = 0$, the result of the vector of probabilities is given by Eq. (1) its $\lambda = \{0.25, 0.25, 0.25, 0.25\}$, which means that all kernels have the same chance to be sampled in each bootstrap model, i.e: maximum diversity between kernel functions. Differently, a large value for β would quickly scale difference between models, consequently, the vector of probabilities λ would accumulate in a single kernel, and just one would be sampled, i.e: minimum diversity since it is the same kernel function for all bootstrap samples returning to the traditional bagging approach.

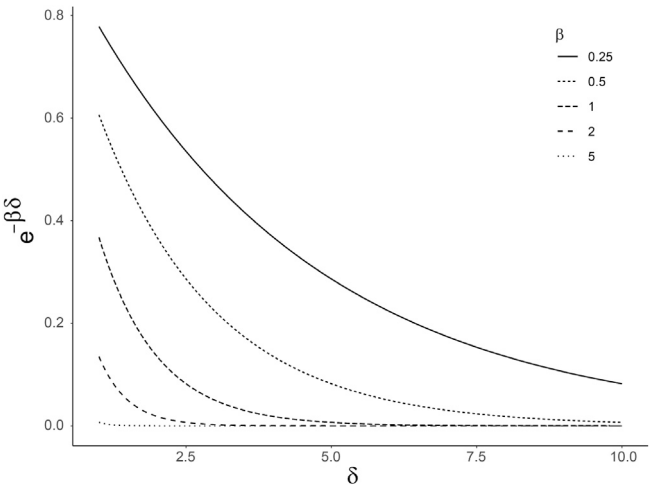

Fig. 1. The correlation coefficient and its relation with the calculation of the probabilities λ. As β increases the penalty given by δ values decrease.

After this, a weight w_i is assigned to each bootstrap model calculated for g_i $\forall i = 1, \dots, B$. The weight is given by Eq. (2).

$$w_i = \frac{e^{-\beta\Lambda_i}}{\sum_{j=1}^{B} e^{-\beta\Lambda_j}}, \quad i = 1, \dots, B, \tag{2}$$

where Λ_i is the Root Mean Square Error of model's prediction g_i using the Out of Bag Samples ($OOBG_i$), obtained from i bootstrap sample $\forall i = 1, \dots, B$, as test set. All this modeling process is summarized in the pseudo-code exposed in Algorithm 1.

$$G(\mathbf{x}_i) = \sum_{j}^{B} w_j g_j(\mathbf{x}_i), \quad i = 1, \dots, N.$$

Algorithm 1 R/andom Machines

Input: Training Data, Test Data, B, Kernel Functions k $= \{k_{LIN}, k_{POL}, k_{RBF}, k_{LAP}\}$
for each Kernel Function r **do**
  Calculate the single SVR_r, $\forall r = 1, \dots, 4$, where r corresponds to each kernel function
Calculate the probabilities λ_r that it will define the chance of one sample of being randomly selected
Generate B bootstrap samples
for b in B **do**
  Model the bootstrap model $g_b(\mathbf{x_i})$ by sampling a kernel function with probability λ_r
  Assign a weight w_b using $OOBG_b$ samples.
Calculate $G(\mathbf{x})$

The entire regression random machines are schematically presented in Fig. 2, where it is designed for all the steps used in all cases presented in this article. It is important to point out that the traditional support vector regression bagging can be interpreted as a particular case of the RRM when the probability of selecting one kernel function is equal to one, and every bootstrap model has the same weight. In

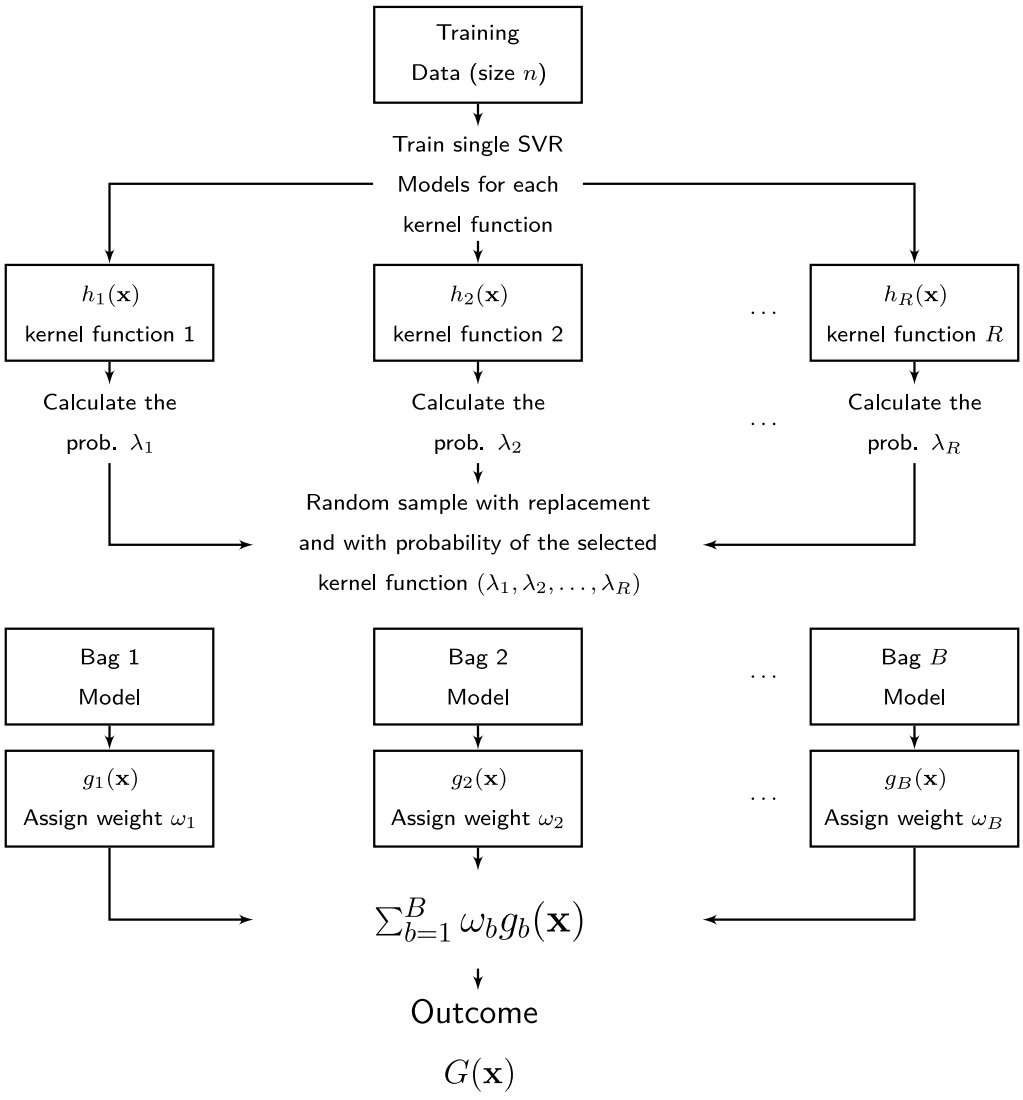

Fig. 2. Workflow followed by the regression random machines.

the same manner, the traditional support vector regression can also be interpreted as a particular case of regression random machines, when a single is selected and $B = 0$.

All the procedures of this paper were performed on a personal laptop with the following configurations: Linux, 64-bit Operating System, Intel Core processor i5-3210M 2.50 GHz and 16 GB of RAM. Moreover, R Software version 3.6.3 with packages *keras* (Chollet, Allaire, et al., 2017) and *kernlab* (Karatzoglou, Smola, Hornik, & Zeileis, 2004). The Regression Random Machines (RM) was also implemented in R language and it can be used through the ***rmachines*** package, under continuous development and available at GitHub (https://github.com/MateusMaiaDS/rmachines).

In general, regression random machines, exposed in this section and summarized in Algorithm 1 and Fig. 2, considers a random mixture of possible kernel functions combined with a properly bagging ensemble which considers the strength and agreement of the single models. At each step of the bagging procedure a random choice of the kernel function is realized, based on a strength of the single models, which decreases the paired agreement of the combined models and improves the general strength of the final model. These concepts are detailed in the next two sections.

3.1. Regression random machines and the bootstrap aggregating

Let $\tau = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ the training set independent and identically distributed over $P = P(\mathbf{X}, Y)$ and a procedure which uses this learning set to form a predictor $\varphi(\mathbf{x}, \tau)$. Let $(\mathbf{x}, y)$ a single observation in τ, $\phi(\mathbf{x}, k)$ the predictor and k a single and fixed kernel function. The model aggregation $\phi_A(.)$ is given by Breiman (1996),

$$\phi_A(\mathbf{x}, k) = E_\tau[\phi(\mathbf{x}, \tau, k)]$$

Then,

$$\begin{aligned} E_\tau\left\{[y - \phi(\mathbf{x}, \tau, k)]^2\right\} &= y^2 - 2yE_\tau[\phi(\mathbf{x}, \tau, k)] + E_\tau\left[\phi^2(\mathbf{x}, \tau, k)\right] \\ &= y^2 - 2y\phi_A(\mathbf{x}, \tau, k) + E_\tau\left[\phi^2(\mathbf{x}, \tau, k)\right] \\ &\geq y^2 - 2y\phi_A(\mathbf{x}, \tau, k) + \phi_A^2(\mathbf{x}, \tau, k) \quad \left(EZ^2 \geq [EZ]^2\right) \\ &= \left[y - \phi_A(\mathbf{x}, \tau, k)\right]^2. \end{aligned}$$

Thus,

$$E_\tau\left\{[y - \phi(\mathbf{x}, \tau, k)]^2\right\} \geq \left[y - \phi_A(\mathbf{x}, \tau, k)\right]^2.$$

Using a both side integration in this inequality at the joint probability distribution P, implies the mean square error of $\phi_A(.)$ is lower than $\phi(.)$. In general, an aggregated predictor is more suitable than a single predictor. This situation depends on

$$\left\{E_\tau[\varphi(\mathbf{x}, \tau, k)]\right\}^2 \leq E_\tau\left[\varphi^2(\mathbf{x}, \tau, k)\right].$$

The instability of $\varphi(.)$ is proved to be important around the aggregated model. If $\varphi(\mathbf{x}, \tau, k)$ does not change too much in τ the two sides will be nearly equal, and aggregation will not help. This is the reason that $\varphi(.)$ with lower instability does not benefit from the aggregation. In general, support vector models are low-biased methods (Valentini & Dietterich, 2004) based on the structural risk minimization principle (SRM) (Vapnik & Chervonenkis, 1974) which balances the model's complexity to prevent the overfitting problem. This concept used in SVM is closely related to the theory of regularization over the trade-off bias and variance (Devi, Kumar, Shankar, et al., 2019). In this sense, many authors have been discouraged from using SVM as a single base learner aggregation (Huang, Chen, Lin, Ke, & Tsai, 2017; Kim et al., 2002; Stork, Ramos, Koch, & Konen, 2015; Ye & Suganthan, 2012).

A way to overcome this situation is to consider κ as a random variable instead of considering k as a user-defined or a tuned selection. Let k a case in a predefined and finite kernel set κ with K possible kernels. The instability now depends on

$$\left\{E_{\kappa\tau}\left[\varphi(\mathbf{x}, \tau, \kappa)\right]\right\}^2 \leq E_{\kappa\tau}\left[\varphi^2(\mathbf{x}, \tau, \kappa)\right].$$

At the same time, the kernel model aggregation is given $\phi_{KA}(\mathbf{x}) = E_{\kappa\tau}\left[\phi(\mathbf{x}, \tau, \kappa)\right]$. So,

$$\begin{aligned} E_{\kappa\tau}\left[\phi(\mathbf{x}, \tau, \kappa)\right] &= E_{\kappa}\left\{E_{\tau}\left[(\mathbf{x}, \tau, \kappa)\,|\kappa\right]\right\} \\ &= \sum_k E_{\tau}\left[(\mathbf{x}, \tau, \kappa)\,|\kappa = k\right] P(\kappa = k) \\ &= \sum_k \phi_A(\mathbf{x}, \tau, k)\,\theta(k), \end{aligned} \tag{3}$$

with $0 < \theta(k) < 1$ the selection probability of the kernel function k, and $\sum_k \theta(k) = 1$. The Eq. (3) implies that $\phi_{KA}(\mathbf{x}, \tau)$ is the aggregation kernel model. Notice that $\phi_{KA}(\mathbf{x}, \tau)$ depends not only on $\mathbf{x}$ but also the underlying probability distribution $P(\mathbf{X}, Y)$ from which τ is drawn which is unknown. Thus, $\phi_{KA} = \phi_{KA}(\mathbf{x}, P)$ and the bagged estimate is not $\varphi_{KA}(\mathbf{x}, P)$, but rather

$$\varphi_{KB}(\mathbf{x}) = \varphi_{KA}\left(\mathbf{x}, P_{\tau}\right)$$

where P_{τ} is the distribution that concentrates mass $1/n$ at each observation $(\mathbf{x_i}, y_i) \in \tau$ and P_{τ} is called the bootstrap approximation of P (Efron & Tibshirani, 1986).

The choice of different kernel functions in a finite kernel space κ guaranteed suitable for the kernel bagging estimate $\varphi_{KB}(\mathbf{x})$. However, very similar kernel functions imply a bagging procedure with no relevant improvements. Otherwise, since the kernel functions consider feature map functions transformations to higher features spaces, it implies that high dimensional $\mathbf{x}$ generates improvements to the random machines. In fact, the probabilities $\theta(k)$ may be estimated in different manners, however, in this paper, we consider the procedure given by Eq. (1).

3.2. Regression random machines out-of-box

There are theoretical reasons why the regression random machines are an ensemble approach that can reduce the generalization error. The random selection of kernel functions works to diversify different functions that belong to a Reproducing Kernel Hilbert Space (RHKS). The goal of this procedure is to diminish the correlation between regression models that constitute the RRM and increase the strength of them since both components result in greater results to bagged classifiers (Breiman, 2001). Obtain these both features is extremely important in the bagging approach, and works that used these strategies, even on different base learners (*i.e*: tree models Breiman, 2001, or kNN classifiers Gul et al., 2018) achieved greater results them the traditional bagged algorithms.

The correlation concept can be defined as a measure of how much models are similar, while the strength of a model relies on how well it correctly predicts an observation. The estimation of a correlation measure can have different approaches. Considering classification models, for instance, a method to estimate the correlation between models is to calculate the area from decision boundaries that overlaps among them (Turney, 1995). Other estimation methods, still considering classification context, are used by Ho (1998), who defines the similarity, through the agreement measure, as the number of observations that are equally labeled with the same class by different models.

In the regression approach, the correlation/similarity estimation between models can be calculated as the mean of the upper triangle from the correlation matrix given in Eq. (4). This correlation measure also can be called as **agreement** value.

$$\Sigma_{corr} = \begin{pmatrix} \rho_{\hat{y}_1,\hat{y}_1} & \rho_{\hat{y}_1,\hat{y}_2} & \cdots & \rho_{\hat{y}_1,\hat{y}_{B-1}} & \rho_{\hat{y}_1,\hat{y}_B} \\ & \rho_{\hat{y}_2,\hat{y}_2} & \cdots & \rho_{\hat{y}_2,\hat{y}_{B-1}} & \rho_{\hat{y}_2,\hat{y}_B} \\ & & \ddots & \vdots & \vdots \\ & & & \rho_{\hat{y}_{B-1},\hat{y}_{B-1}} & \rho_{\hat{y}_{B-1},\hat{y}_B} \\ & & & & \rho_{\hat{y}_B,\hat{y}_B} \end{pmatrix}_{B\times B} \tag{4}$$

The values of $\rho_{\hat{y}_i,\hat{y}_j}$ are calculated by

$$\rho_{i,j} = \frac{\sum_{k=1}^{T}(\hat{y}_{i_k} - \overline{\hat{y}_i})(\hat{y}_{j_k} - \overline{\hat{y}_j})}{\sqrt{\sum_{k=1}^{T}(\hat{y}_{i_k} - \overline{\hat{y}_i})}\sqrt{\sum_{k=1}^{T}(\hat{y}_{j_k} - \overline{\hat{y}_j})}}. \tag{5}$$

for all $i \neq j = 1, \ldots, B$, and $\hat{y}_i$ it is the vector of predictions from observations that belong to the test set. The strength in this article will be estimated using the Error Score since it captures the prediction performance and has the same range of the correlation measure. This index formulation is given by

$$ES_i = \frac{\varepsilon_i - \varepsilon_{min}}{\varepsilon_{max} - \varepsilon_{min}}, \tag{6}$$

where the $\boldsymbol{\varepsilon}$ is just the RMSE vector over test observations for all methods, and where ε_i is the individual value for that technique $\forall i = 1, \ldots, R$. For instance, suppose three algorithms: regression random machines, support vector linear regression (SVR.Lin) and bagging support vector linear regression (BSVR.Lin), then, after calculating the RMSE for each of them over a test set, the vector $\boldsymbol{\varepsilon} = \{0.1, 0.5, 0.3\}$ is obtained, with the coordinates for each kernel function respectively. Thus, the $\boldsymbol{ES}$ vector is given by $\boldsymbol{ES} = \{0, 1, 0.5\}$ which means that $ES_1 = 0$ was the Error Score for RRM, $ES_2 = 1$ for SVR.Lin and $ES_2 = 0.5$ for BSVR.Lin. As the ES is directly proportional to the Root Mean Square Error, it also can be considered a strength measure. Smaller values of RMSE produced by a regression model imply a stronger model.

To evaluate the correlation and strength of the RRM in comparison with the traditional bagged version of SVR, the algorithm was applied to overall models of simulated data detailed at Section 4. We look for the model which has the lowest RMSE and Error Score (i.e: greater strength) **and** the lowest correlation measure. A model with a small agreement can benefit more from the bagging procedure (Breiman, 2001). However, just small values of correlation are not enough, since this lower value can represent a weak model, i.e, a model which is not capable of predicting new observations well. The result is summarized in Table 2. Both RMSE and Agreement were calculated using a 30 Repeated Holdout validation set with a split ratio of 70%–30% training-test. The parameters of the methods were: B=100, $\gamma = 1$, C=1, $\beta = 2$ and $\epsilon = 0.1$.

The strength of the models is affected by the agreement and vice-versa, so optimizing both measures at the same time is a difficult effort. The relation between them can be analyzed in Table 2. Observing simultaneously the RMSE and the Agreement measure from traditional bagging approaches exists a trade-off between them. Considering Table 2, if the RMSE is the lowest among them, its agreement is the highest. This trade-off is minimized in the RRM case, which, despite presenting the lower RMSE in most of the cases, is not reflected in the highest agreement measure among all methods. Therefore, regression random machines have a low correlation and great strength, desirable features to produce a good bagging approach.

The idea of how the random selection of kernel functions can increase the diversity of regression random machines model when

Table 2
Summary of Strength (RMSE) and Agreement over simulated data.

Model	n	BSVR.Lin		BSVR.Pol		BSVR.Gau		BSVR.Lap		RRM	
		RMSE	AGR	RMSE	AGR	RMSE	AGR	RMSE	AGR	RMSE	AGR
1	30	0.1189	0.6427	0.0812	0.2417	0.0773	0.6146	0.0777	0.7342	0.0725	0.5236
	100	0.0538	0.9531	0.0216	0.3337	0.0196	0.7873	0.0246	0.9163	0.0147	0.8806
	1000	0.0117	0.9953	0.0049	0.5841	0.0028	0.9187	0.0038	0.9797	0.0030	0.8822
2	30	0.2972	0.5300	0.2388	0.4839	0.2209	0.8472	0.2229	0.8329	0.1913	0.7263
	100	0.1356	0.2617	0.1086	0.9340	0.1361	0.9620	0.1394	0.9648	0.1287	0.9303
	1000	0.0272	0.1630	0.0214	0.9926	0.0292	0.9954	0.0284	0.9916	0.0213	0.9276
3	30	0.1788	0.5754	0.5976	0.4217	0.2672	0.5810	0.2644	0.7254	0.1983	0.4719
	100	0.0641	0.8014	0.1715	0.1746	0.1260	0.6121	0.1183	0.8669	0.0749	0.6634
	1000	0.0279	0.9821	0.0443	0.5348	0.0253	0.8171	0.0245	0.9642	0.0252	0.8622
4	30	0.5890	0.7763	1.4439	0.2917	0.6318	0.7119	0.6273	0.7904	0.4998	0.6229
	100	0.1671	0.9142	0.4010	0.5741	0.3106	0.8236	0.2577	0.9065	0.2199	0.7995
	1000	0.0608	0.9927	0.1172	0.8581	0.0551	0.9700	0.0408	0.9833	0.0515	0.9223
5	30	0.2044	0.5607	0.2242	0.2609	0.1664	0.5638	0.1647	0.7267	0.1630	0.4967
	100	0.0811	0.8381	0.1487	0.3304	0.0956	0.7468	0.0958	0.8854	0.0743	0.7168
	1000	0.0215	0.9854	0.0333	0.8492	0.0267	0.9249	0.0242	0.9788	0.0225	0.9141
6	30	0.1813	0.6427	0.3004	0.2417	0.1397	0.6146	0.1411	0.7342	0.1373	0.5236
	100	0.0641	0.9531	0.1553	0.3337	0.1219	0.7873	0.1181	0.9163	0.0655	0.8806
	1000	0.0206	0.9953	0.0429	0.5841	0.0289	0.9187	0.0239	0.9797	0.0226	0.8822
7	30	0.1813	0.8227	0.3004	0.2767	0.1397	0.6822	0.1411	0.7825	0.1373	0.6855
	100	0.0641	0.9774	0.1553	0.4250	0.1219	0.8694	0.1181	0.9516	0.0655	0.8941
	1000	0.0206	0.9974	0.0429	0.6676	0.0289	0.9539	0.0239	0.9852	0.0226	0.9113
8	30	0.3852	0.3062	0.5018	0.3884	0.4003	0.5006	0.4072	0.6054	0.4057	0.3633
	100	0.3463	0.3791	0.3931	0.3763	0.3732	0.6027	0.3760	0.6778	0.3612	0.4037
	1000	0.1405	0.4648	0.1593	0.9603	0.1609	0.7719	0.1511	0.9143	0.1496	0.8568
Total	–	3.4430	17.5107	5.7094	12.1191	3.7061	18.5776	3.6149	20.7939	3.1283	17.7414

compared with the traditional SVR bagging algorithm, cannot be so clear at the first moment. In order to observe clearly how it works, we examine **Model 1** in Section 4.1, with $n = 1000$, as a study case to see graphically the modeling process. Fig. 3 shows the level curves from the true data generation surface, the predicted hyperplane for RRM, and each single bootstrap model of a single kernel function. It is notable that for each different kernel function the regression surface is distinct, leading to diversity and subsequent reduction of correlation between models. Moreover, it is possible to notice that the regression random machines surface, built through the combination of these single models, is the one that is closer to the real data generation hyperplane, reinforcing that it works as the best model in that case.

The same behavior was observed on the 26 real data sets, presented in Section 4, where the agreement is also calculated and compared with the strength (Error Score) of each model (Fig. 4). Although the low values of generalization error from BSVR.Lin and BSVR.Lap they present large agreement values. In opposition, despite the low values of agreement from BSVR.Pol and BSVR.Gau they produce great values of Error Score. The unique method that can perform the optimal values for both is the regression random machines.

3.3. The correlation coefficient β

Another way to study the correlation-strength trade-off in the RRM procedure is through the coefficient β presented in the Eqs. (1)–(2). As mentioned before, the β coefficient can calibrate the diversity of kernel functions used during the bagging procedure. If we consider $\beta = 0$ the RRM will hold that all kernel functions will be sampled and weighed equally. On the other hand, if we use greater values of β the RRM's behavior will be close to traditional SVR bagging, since just a single kernel function will be chosen.

In order to demonstrate this performance, we evaluate the standardized RMSE and Agreement on three benchmarks datasets which will be used again in Section 4. We considered the values of β in a grid that range from 0 to 5, with the length of 21 intervals. Both measures were calculated in a holdout validation with split ratio of 70%–30%, and setting the parameters $B = 100$, $\gamma = 1$ and $d = 2$. The results are summarized in three Data Sets: *Taiwan*, *Boston Housing* and *Friedman #1* presented in Fig. 5.

Through different values of RMSE and Agreement, all of them present the same behavior: for small values of β, i.e: minimum correlation between models, we have a weaker (represented through the large RMSE values) model from the RRM algorithm. As the beta value increases the weighting on the kernel functions and bootstrap models predictions are applied increasing the agreement and reducing the RMSE. However, at some inflection point, the RMSE starts to increase and the agreement continues to grow. The reason for this result can be explained by Breiman (1996), where it is defined that stable base-models, as SVR models, in this case, will not benefit from the aggregation procedure, and may even depreciate the model. Large values of β start to penalize the RRM in a way that just one, or few different, kernel functions will be chosen, and as SVR (Breiman et al., 1996) is defined as a stable model (i.e: bootstrap replications produces small changes in the model) this may lead to worst results. Therefore, the key to the improvement from regression random machines is to add the instability in SVR, necessary for bagging procedures (Breiman, 1996), through the random sample of kernel functions and the weights associated with the predictions.

The β also can be defined as a hyperparameter of the model and can be tuned in order to achieve the lowest generalization error. From some empirical results, in this article, the default choice of β was 2, since in most cases the best choice has been around this value and there was not much improvement from the grid search procedure for this parameter.

4. Applications on artificial and real data

The proposed method was evaluated on simulated and real data scenarios; The most diverse behaviors were used to exploit a wide range of variety.

4.1. Artificial data application

Different scenarios were used to study the regression random machines on simulated data. The objective was to evaluate the

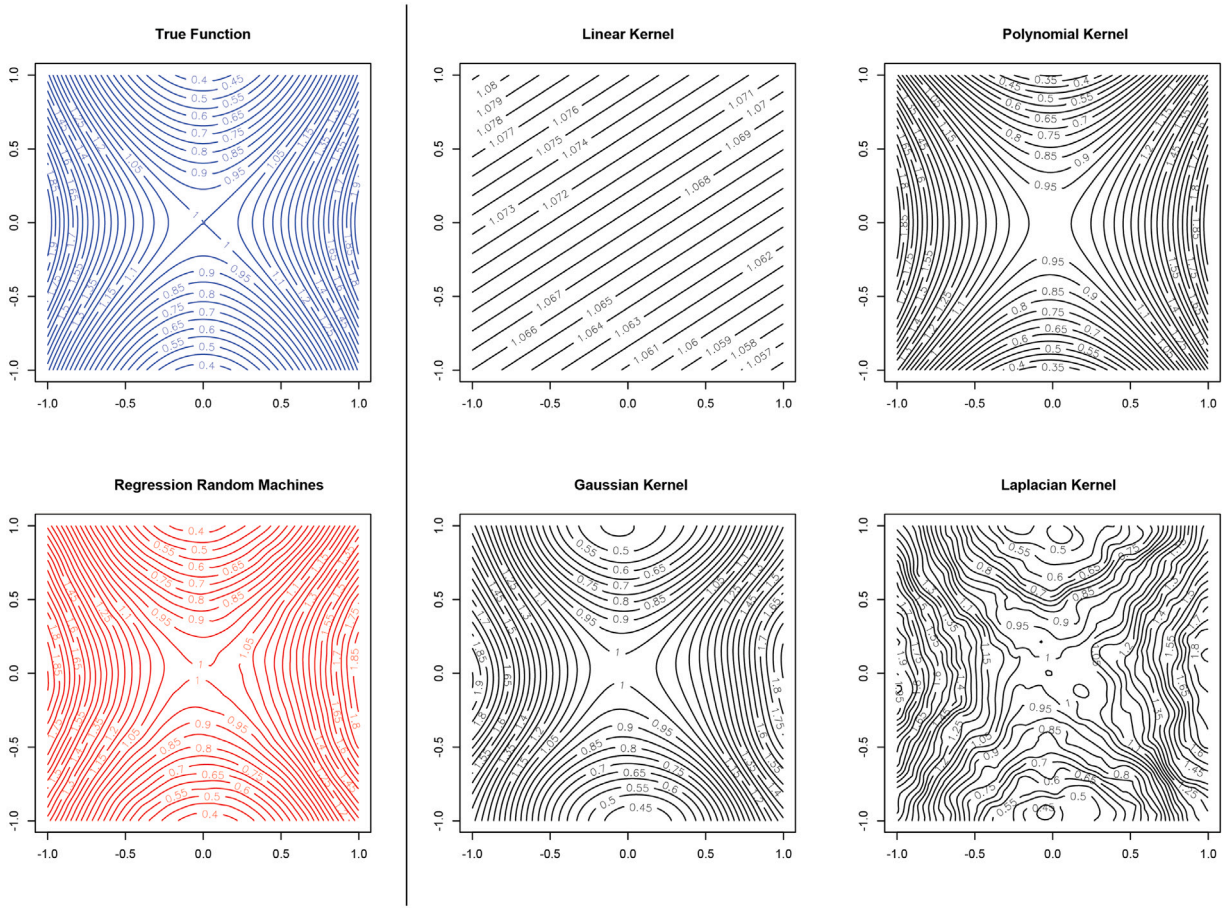

Fig. 3. Level Curves for different single Kernel Function models used in the regression random machines. It is important to analyze the diversity between the different kernel functions, and how the RRM surface is the closest to the True Function.

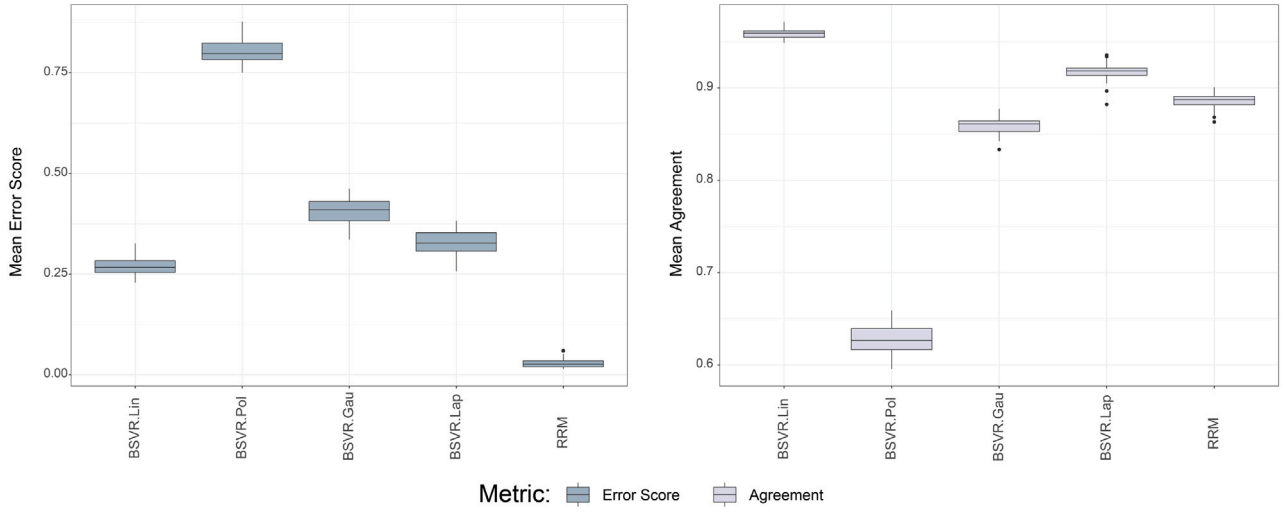

Fig. 4. Boxplots of the mean Error Score and Mean Agreement for each method.

performance and behavior of the model when we have a controlled experiment. Eight different data sets generation scenarios were tested. The **Models 1-5** are toy examples and can be found in Scornet (2016), the **Model 6** in Van der Laan, Polley, and Hubbard (2007), the **Model 7** in Meier, Van de Geer, Bühlmann, et al. (2009) and the **Model 8** is presented in Roy and Larocque (2012). The simulations from **1-7** has the vector of independent predictions $\boldsymbol{X} = (X_1, \ldots, X_p)$ and $\boldsymbol{X}$ follow a uniform distribution $[0, 1]^p$. In the **Model 8** each predictor X_i follow a standard normal distribution. Also, we define the transformation $\tilde{X}_i = 2(X - 0.5), \quad i = 1, \ldots, p$. For each case the sample size changed among the values of $n = \{30, 100, 1000\}$. All the scenarios are described below:

- **Model 1**: p=2, $Y = \tilde{X_1^2} + e^{-\tilde{X_2^2}} + \mathcal{N}(0, 0.25)$
- **Model 2**: p=8, $Y = \tilde{X_1}\tilde{X_2} + \tilde{X_3^2} - \tilde{X_4}\tilde{X_7} + \tilde{X_5}\tilde{X_8} - \tilde{X_6^2} + \mathcal{N}(0, 0.5)$
- **Model 3**: p=4, $Y = -sin(\tilde{X_1}) + \tilde{X_2^2} + \tilde{X_3} - e^{-\tilde{X_4^2}} + \mathcal{N}(0, 0.5)$

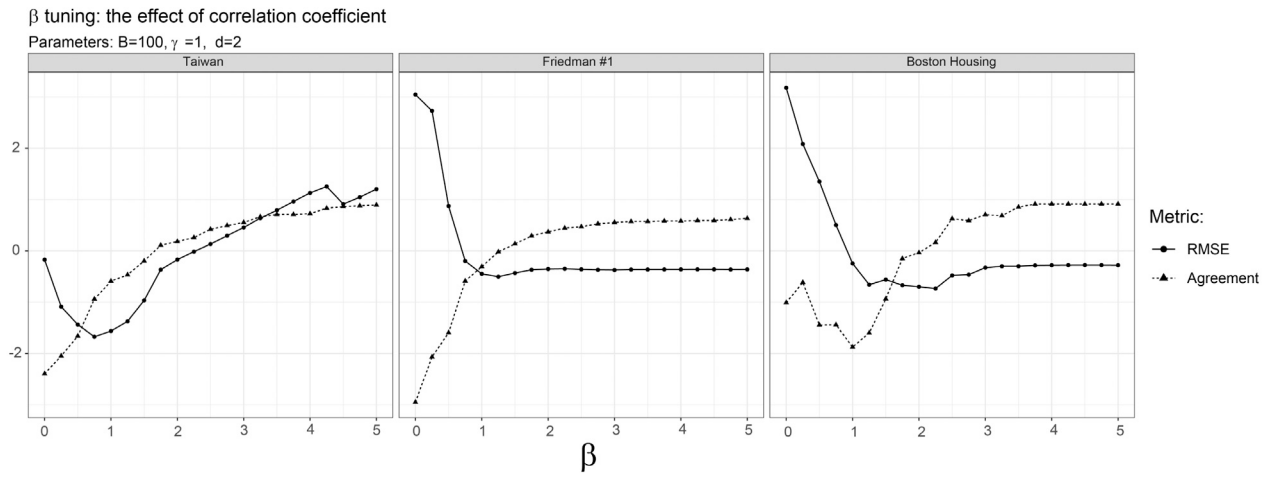

Fig. 5. Standardized values of RMSE and Agreement for the different values of β.

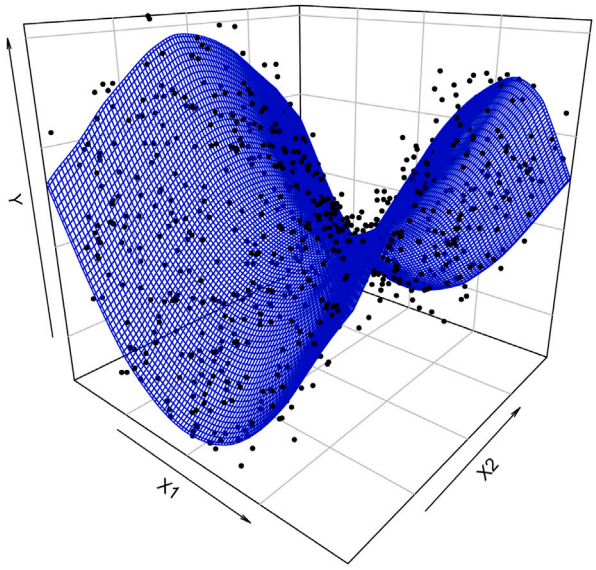

Fig. 6. Regression hypertube of Regression Random Machine algorithm.

- **Model 4**: p=4, $Y = -\tilde{X_1} + (2\tilde{X_2} - 1)^2 + 2sin(2\pi\tilde{X_3})/(2 - sin(2\pi\tilde{X_3})) + sin(2\pi\tilde{X_4}) + 2cos(2\pi\tilde{X_4}) + 3sin^2(2\pi\tilde{X_4}) + 4cos^2(2\pi\tilde{X_4}) + \mathcal{N}(0, 0.5)$
- **Model 5**: p=8, $Y = \mathbb{1}_{\tilde{X_1}>0} + \tilde{X_2^3} + \mathbb{1}_{\tilde{X_3}+\tilde{X_4}-\tilde{X_6}-\tilde{X_5}>1+\tilde{X_7}} + e^{-\tilde{X_8^2}} + \mathcal{N}(0, 0.5)$
- **Model 6**: p=6, $Y = \tilde{X_1^2} + \tilde{X_2^2}\tilde{X_3}e^{-|\tilde{X_4}|} + \tilde{X_6} - \tilde{X_5} + \mathcal{N}(0, 0.5)$
- **Model 7**: p=4, $Y = \tilde{X_1} + 3\tilde{X_2^2} - 2e^{-\tilde{X_3}} + \tilde{X_4}$
- **Model 8**: p=6, $Y = X_1 + 0.707X_2^2 + 2\mathbb{1}_{X_3>0} + 0.873\log(X_1)|X_3| + 0.894X_2X_4 + 2\mathbb{1}_{X_5>0} + 0.464e^{X_6} + \mathcal{N}(0, 1)$

An illustration about how the regression hyperplane created by the RRM can be seen in Fig. 6 which the **Model 1** of data generation is used as example.

The repeated holdout with 30 repetitions was used as a validation technique with a split ratio of a training-test of $70\% - 30\%$. The default parameters of models were $\gamma = 1$, $B = 100$, $C = 1$, $\beta = 2$, $d = 2$ and $\epsilon = 0.1$. The value $\beta = 2$ proved to be a stable choice over the experiments in Section 3. The other parameters are considered default as SVR standard methodology. Table 3 summarizes the result, and from it is possible to realize that the Regression Random Machine outperformed the other methods in the majority of simulations setups that were presented and compared with support vector regression with polynomial kernel function (SVR.Pol), support vector regression with Gaussian kernel (SVR.Gau), support vector regression with Laplacian kernel (SVR.Lap), bagging SVR.Pol (BSVR.Pol), bagging SVR.Gau (BSVR.Gau) and bagging SVR.Lap (BSVR.Lap). The bold RMSE mean values indicate the lowest among all considered methods. In many of the considered model scenarios we can notice that bagging procedure does not improve its single SVR. The support vector regression with Laplacian kernel had a satisfactory predictive performance in single and bagging versions. The single polynomial kernel had the worst general predictive performance. Regression random machines had the best general predictive performance, minimum value or close to the minimum to all the scenarios.

4.2. Study of tuning process

The regression random machines presents robustness concerning the hyperparameter selection for the SVR base models avoiding the computational cost of performing a grid search to select the best model setting, without harm the predictive performance. To evaluate this feature, all the simulated scenarios were replicated varying the sample size in $n = \{30, 100, 300, 500, 1000\}$ with a 30 repeated holdout cross-validation setting using a split ratio of training-test of 70%–30%. To perform a complete grid search for the single SVR models shifting among the four kernel functions: linear, polynomial, Gaussian and Laplacian. For the other hyperparameters the range of $\gamma = C = \{2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3\}$ and $\epsilon = \{0.001, 0.01, 0.1\}$. The regression RM was applied without any tuning, and with the default hyperparameters also described in Section 4.1 . To measure the predictive performance three metrics were used: the RMSE, already defined in the article, the Mean Absolute Error ($MAE = \frac{1}{n}\|\mathbf{y} - \hat{\mathbf{y}}\|$) and the Cosine Similarity ($COS = \frac{\mathbf{y}\cdot\hat{\mathbf{y}}}{\|\mathbf{y}\|\times\|\hat{\mathbf{y}}\|}$). The $RMSE = 0$ and $MAE = 0$ represent a perfect prediction scenario while the $COS = 1$ represent the optimal prediction result. The results are summarized in Tables 4 and 5, where the average values of RMSE, MAE and COS are shown. The values for each row corresponds to a tuning over the SVR.LIN, SVR.POL, SVR.RBF, SVR.LAP and a default version of regression RM.

The results show that, in general, RM presents an equivalent performance when compared with the SVR with the best kernel and best hyperparameters selected by the tuning process. In general, the RM can outperform SVR or present the worst result when compared with the best SVR, however, in general, the proposed ensemble model can yield reliable predictions with a smaller time processing.

Fig. 9 highlights the time difference between the computational cost of realizing a tuning in the SVR models and applying the RM. The average time in minutes of the tuning process can be almost ten times

Table 3
Summary of the simulation's results for the different databases. Each value corresponds to the mean RMSE method on the test data . In bold the lower value of the generalization error.

Model	n	SVR.Lin	SVR.Pol	SVR.Gau	SVR.Lap	BSVR.Lin	BSVR.Pol	BSVR.Gau	BSVR.Lap	RRM
1	30	0.4443	0.3328	0.2189	0.2604	0.4305	0.2942	0.2454	0.2882	**0.2134**
	100	0.4507	0.1403	0.1482	0.1631	0.4452	0.1370	0.1522	0.1770	**0.1226**
	1000	0.3877	0.1110	0.1091	0.1132	0.3876	0.1108	0.1088	0.1120	**0.1069**
2	30	1.2274	1.4203	0.9653	0.9435	1.2221	**0.9146**	0.9643	0.9511	0.9191
	100	1.0425	0.7983	0.9333	0.9016	1.0293	0.7401	0.9379	0.9122	**0.7390**
	1000	0.8900	0.4998	0.7996	0.6643	0.8888	0.4983	0.8176	0.6937	**0.4980**
3	30	0.9202	2.1038	1.0702	1.0368	0.9181	1.5041	1.1493	1.1121	**0.8761**
	100	0.6176	1.4123	0.8254	0.7077	0.6116	1.3375	0.8906	0.7738	**0.5959**
	1000	0.6086	1.2475	0.5675	0.5373	0.6082	1.2220	0.5612	0.5338	**0.5334**
4	30	2.2237	4.4928	2.7105	2.6977	2.3196	3.8646	2.8099	2.7963	**2.1556**
	100	2.2462	2.9495	2.3342	2.1906	2.2313	2.8389	2.3770	2.2504	**2.1394**
	1000	2.1302	3.0750	**1.0068**	1.1613	2.1295	3.0458	1.0734	1.2927	1.0970
5	30	0.9664	1.8465	1.1168	1.0756	**0.8867**	1.3307	1.1088	1.0846	0.9025
	100	0.7740	1.9348	1.0089	0.9074	**0.7700**	1.4642	1.0115	0.9355	0.7757
	1000	0.7003	0.9621	0.8835	0.6925	0.6998	0.9561	0.8978	0.7115	**0.6876**
6	30	**0.6806**	2.7753	0.9458	0.8629	0.7353	1.4283	0.9517	0.8968	0.7554
	100	0.6380	1.2054	0.9416	0.7962	0.6395	1.1986	0.9687	0.8440	**0.6353**
	1000	0.5792	1.0204	0.6521	0.5491	0.5792	1.0015	0.6709	0.5570	**0.5453**
7	30	**0.6806**	2.7753	0.9458	0.8629	0.7353	1.4283	0.9517	0.8968	0.7554
	100	0.6380	1.2054	0.9416	0.7962	0.6395	1.1986	0.9687	0.8440	**0.6353**
	1000	0.5792	1.0204	0.6521	0.5491	0.5792	1.0015	0.6709	0.5570	**0.5453**
8	30	2.2623	4.2416	2.2105	2.0928	2.0908	2.4499	2.2323	2.1550	**2.0283**
	100	1.8324	2.5624	2.0880	1.9245	1.8286	2.3039	2.1121	1.9658	**1.7994**
	1000	1.9270	2.1882	1.8161	**1.4392**	1.9252	2.1468	1.8884	1.5254	1.5107
Total	––	25.4471	42.3213	25.8916	23.9261	25.3307	34.4164	26.5210	24.8669	**21.5729**

greater than the default RM for a sample of 30 observations. As the sample size increases, this ratio between the SVR. Tuning time and RM increase as well, showing that the RM can avoid tuning processes that would be computationally infeasible for large datasets.

To analyze the marginal effects of the tuning process, two other experiments were performed. The first one is show in Fig. 7, where for each simulated scenario the ϵ was kept fixed while all combinations of $C = \{2^{-2}, 2^{-1}, 1, 2\}$ and $\gamma = \{2^{-2}, 2^{-1}, 1, 2\}$ using a Gaussian Kernel for the single SVR model. The RM model was run with the default parameters, and with $B = \{25, 100\}$. The results enhance the previous interpretation about the effectiveness of RM to yield good predictions without any tuning processes. With exception of Scenario 4, all other hyperparameters combination of the SVR.RBF presented greater values of RMSE than the default versions of RM.

The second one is show in Fig. 8, where, using a RBF kernel function, the same setting for $C = \{2^{-2}, 2^{-1}, 1, 2\}$ and $\gamma = \{2^{-2}, 2^{-1}, 1, 2\}$ was used, in addition with the variation over $\varepsilon = \{0.1, 0.01, 0.001\}$. Besides the RBF kernel, the polynomial with degree $d = \{1, 2, 3\}$ was also used as an example of hyperparameter tuning. In this experiment, instead of applying all possible combinations, each parameter was changed while the others were kept constant on their default values. The result was evaluated over the simulated Model 2, 4 and 8 respectively. The settings for RM remain the same as the previous experiment. Fig. 8 shows, that except the variation of the cost parameter C in Model 4, RM presented the lowest RMSE value indicating that their simplest setting surpass a single tuning of SVR hyperparameters with a lower computational cost (see Table 6).

4.3. Real data application

The methodology was applied on 26 real-world datasets from the UCI Repository (Dua & Graff, 2017) to evaluate its performance. The datasets present a wide variety in the number of observations, dimensionality, and type of data, all concerning regression tasks. The UCI benchmarkings are extensively used in the machine learning literature as a resource of comparison for many algorithms for real-world applications due to their variety and reliability (Amarnath, Balamurugan, & Alias, 2016; Khan, Arif, Siddique, & Oishe, 2018) — having more than 4,900 citations at Google Scholar. Besides the UCI datasets, some of them were gathered from the package *MASS* (Ripley et al., 2013), and their description can be found in the documentation of the package. Table 7 summarizes all datasets considered. The continuous features were scaled to zero mean and unit variance, except for the discrete features which went through a one-hot-encoding process. The validation technique used was the repeated holdout with 30 repetitions and a split ratio of training-test of $70\% - 30\%$.

The regression random machines were compared with the bagged SVR approach using each one of single kernel functions shown in Table 1, and with the standard SVR procedure with the same kernel functions. The chosen parameters were: the parameter $\epsilon = 1$, the cost parameter $C = 1$, the number of bootstrap samples $B = 100$, the degree of polynomial kernel $d = 2$, and the hyperparameter γ from the Table 1, $\gamma = 1$. The result is resumed in Fig. 10 considering the Root Mean Squared Error (RMSE).

As demonstrated in Fig. 10, the RRM shows lower generalization error than the other bagged support vectors using unique kernel functions. Comparing the RRM with the traditional bagged SVR, it outperforms almost 90.9% of times the Kernel Linear Bagging, 96.9% for the Kernel Polynomial Bagging, 97.2% for the Gaussian Bagging, and 94.7% for the Laplacian Kernel Bagging. These results show off that the random weighted choice of the functions of the kernel reduced, mostly, the error from the predicted values. The difference is also present when the regression random machines are compared with the singular SVR, where the RM is more accurate 91.2% of times considering the Kernel Linear, 96.4% for the Kernel Polynomial, 94.6% for the Gaussian Bagging, and 84.5% for the Laplacian Kernel. Fig. 11 shows boxplots for the mean values for the Error Score, presented in Section 3.1, overall the 30 holdout repetitions for all 26 benchmarking data sets. Analyzing the results it is clear to see that the RRM has, generally, good performance when compared with the traditional methods. This approach also deals with the problem of the choice of the best kernel

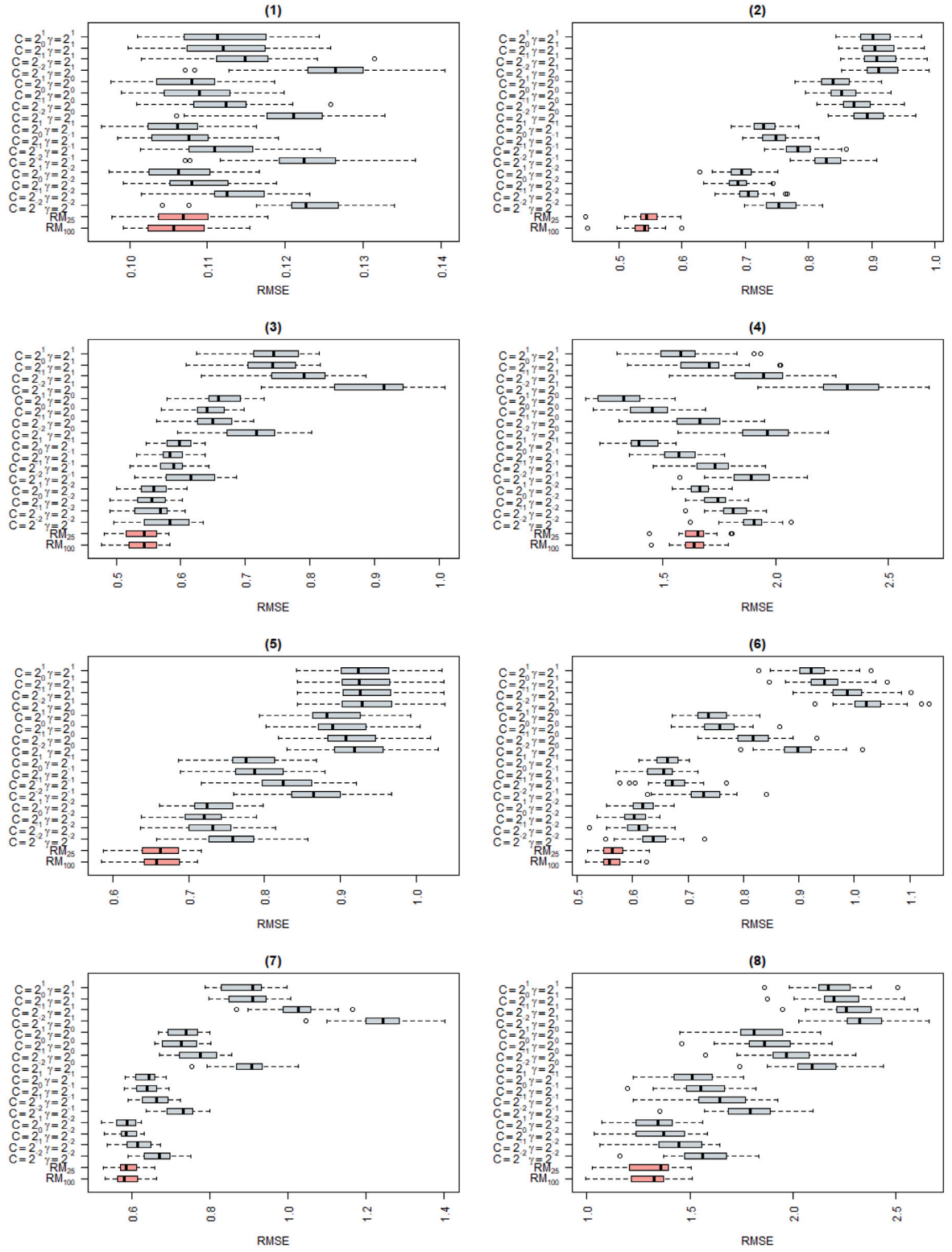

Fig. 7. Performance capacity for the tuning processes of hyperparameters C and *gamma* for SVR with Gaussian kernel over all scenarios from Model 1 to Model 8. The validation considers $n = 1000$ and a repeated holdout 70-30 with thirty repetitions.

function, since it is not necessary to perform a grid-search among all the different functions and define which one has a lower test error. For this reason, the RRM algorithm can be considered efficient, as it can reduce the prediction error and the computational cost.

As hyperparameter tuning is an important procedure in the support vector machine regression algorithm (Duan, Keerthi, & Poo, 2003), the value of γ was changed in order to study how its variation affects the behavior of RRM. The setting of the parameters was the set of values $\gamma = \{2^{-3}, 2^{-2}, 2^{-1}, 2^{0}, 2^{1}, 2^{2}, 2^{3}\}$ over the same data sets. The result is shown in Fig. 12, where it is possible to see that the RRM surpassed the other bagging and single models. As said before the selection of these parameters, as the kernel function, has a direct impact on the model

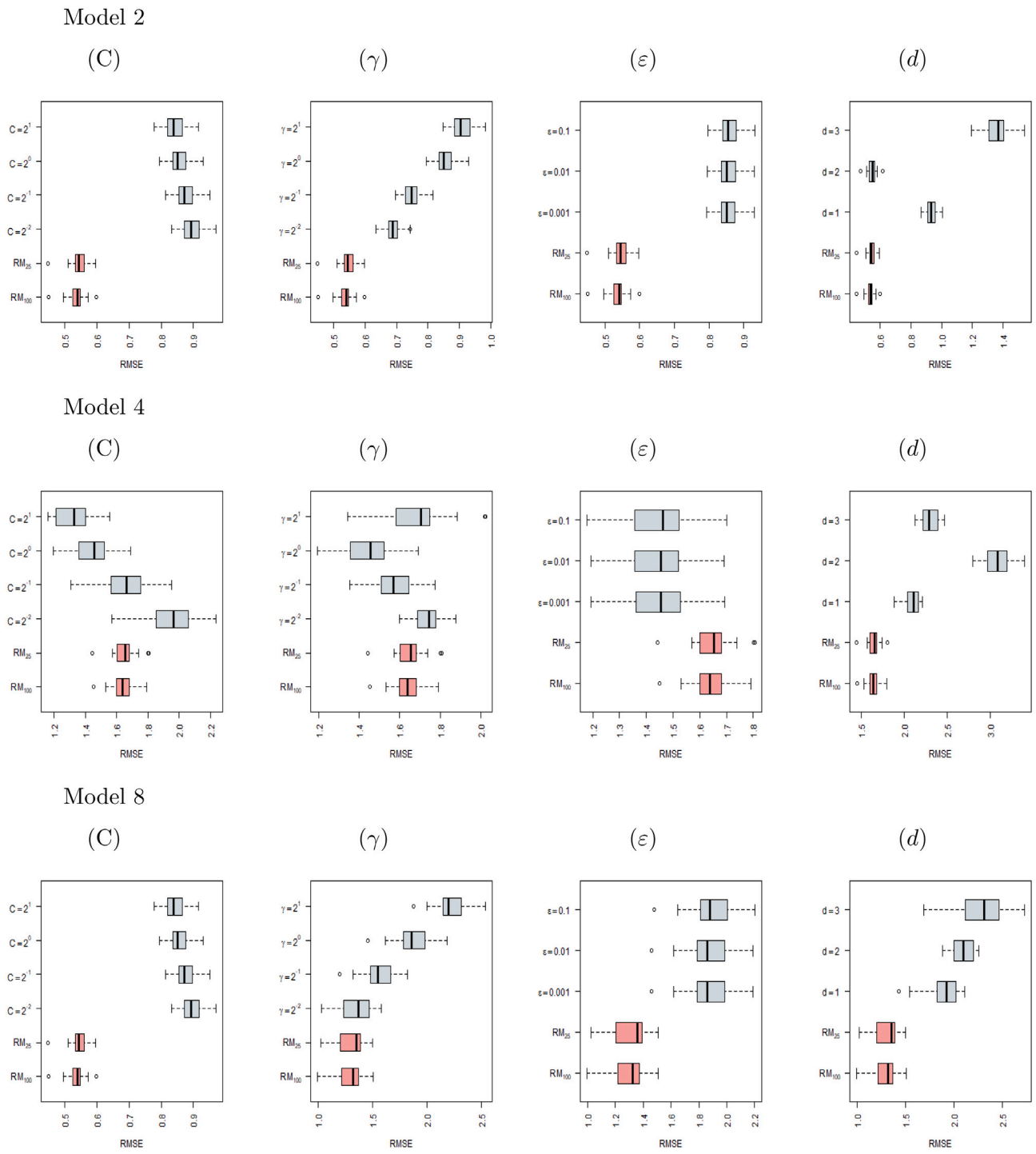

Fig. 8. Marginal effects of tuning a SVR.RBF model varying the C, γ and ϵ, and SVR.Pol varying the degree d compared with the default values of RM over the simulated Models 2, 4 and 8 respectively.

performance, and the results fortify the supposition that RRM gives a good and consistent result for a wide range γ values.

4.4. Bolsa familia programme application

Several governments around the world seek to combat inequality and poverty through social programs. This is the case of the Brazilian Bolsa Familia Programme, which was created in 2003 with the objective of promoting poverty reduction in Brazil. The dataset contains Brazilian municipalities ten covariates and the rate of people using Bolsa Familia as response variable. This rate, which is the variable of interest in the study, is defined by the number of people who receive the aid divided by the total population of the municipality. The importance of predicting with a high level of accuracy the rate of use of the program can help the government to better manage resources, in addition to understanding which regions of Brazil are suffering more with social inequality. The source of this dataset may be found in Paz et al. (2020) and Maia, Azevedo, and Ara (2021) and can be accessed in *rmachines* package in R Language.

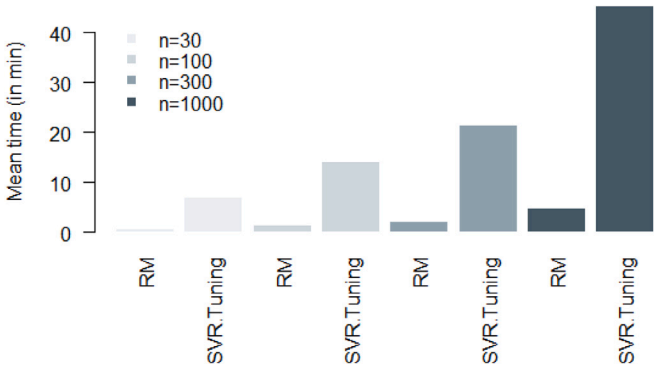

Fig. 9. Mean time for the SVR tuning and regression RM for all eight scenarios and sample size $n = \{30, 100, 300, 1000\}$.

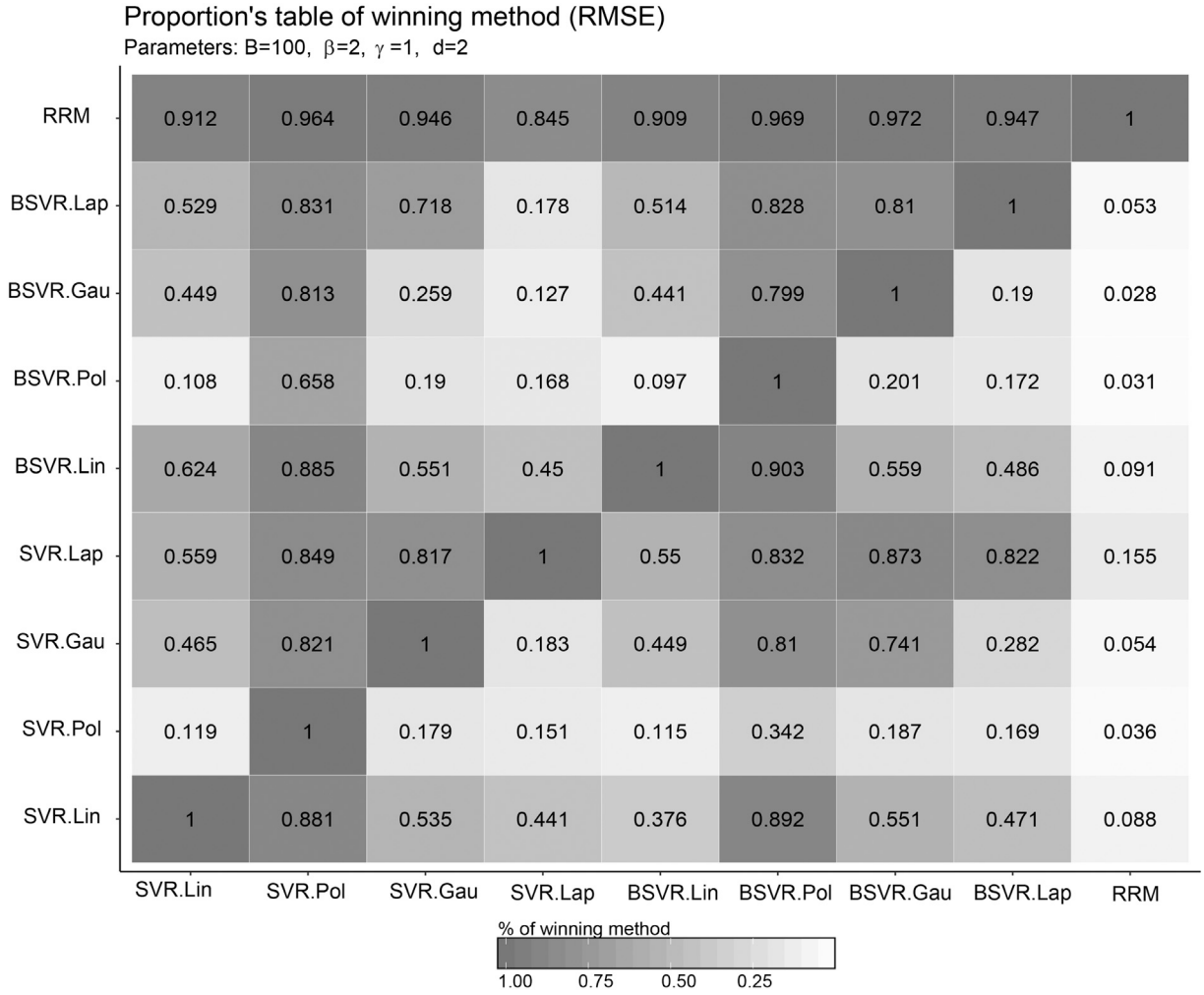

Fig. 10. The proportion of the number of times which a method obtained lower RMSE than the others. The proportion summarizes the applications overall 26 datasets and 30 holdout values.

The applied machine learning methods are composed by regression random machines, extreme gradient boosting (Friedman, 2001), light gradient boosting machine (Ke et al., 2017), both with 100 training rounds, support vector regression models using linear ($C = 1$ and $\lambda = 1$), polynomial ($C = 1$ and $d = 2$) and Gaussian kernels ($C = 1$ and auto-tuning for λ) as well as dense multi-layer perceptions in a deep learning context (Hinton et al., 2012). The deep learning multi-layer perceptron (DP-MLP) was considered for two different architectures with the following hidden layers: (1) 128-64-32 and (2) 256-128-64. The activation function was the scaled exponential linear unit (SeLU) and adam optimizer using default learning rate 0.001 (Klambauer, Unterthiner, Mayr, & Hochreiter, 2017).

The predictive performance of the models was measured using the root mean squared error (RMSE), mean absolute error (MAE) and cosine similarity (COS) metrics. These measures were calculated in a repeated holdout validation technique with 30 repetition by 75%–25%

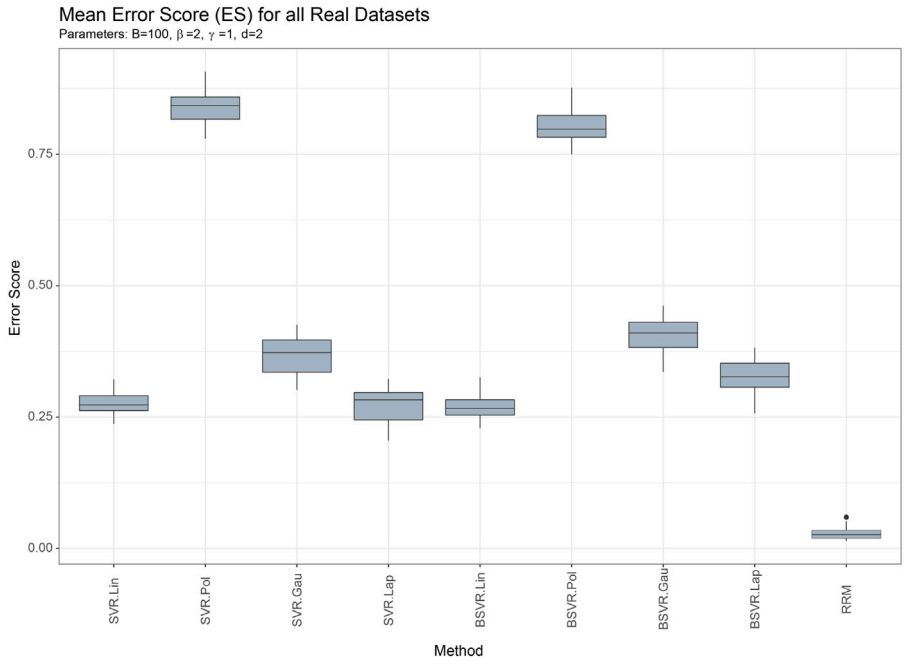

Fig. 11. Boxplot of the mean values for each technique used.

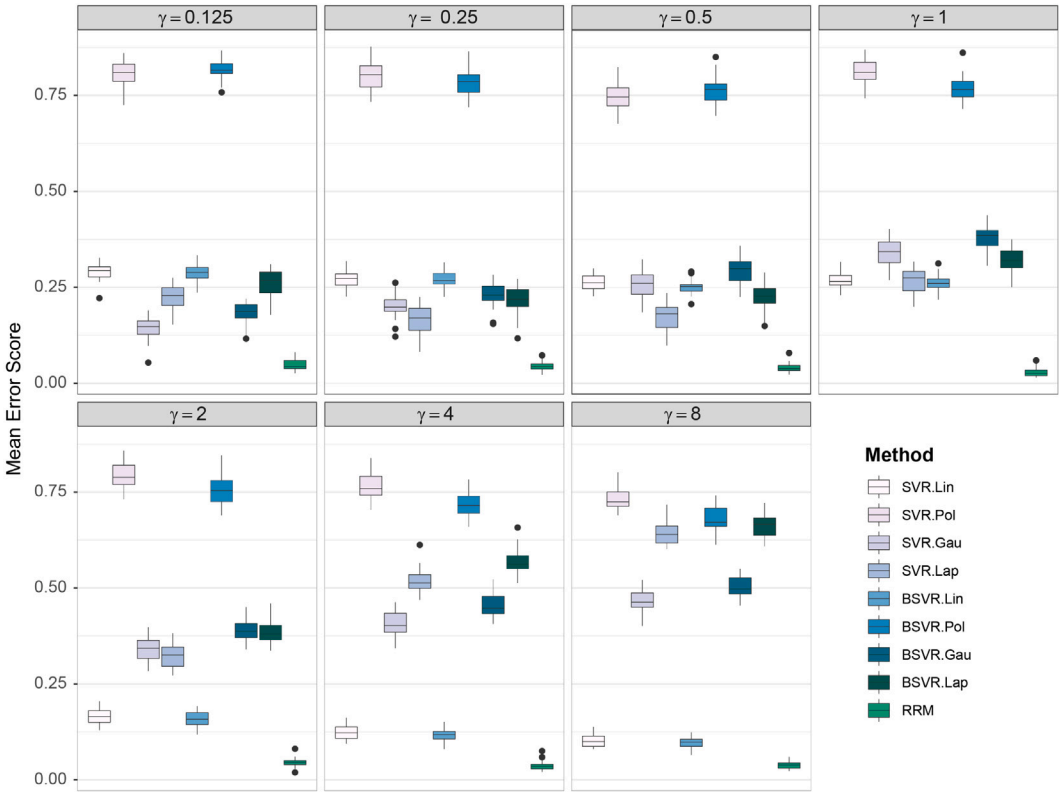

Fig. 12. Summary of the mean values for the Error Score applied over 26 real datasets with the different kernel functions and gamma's values.

for training and test data, respectively. Table 8 displays the results of predictive performance metrics. Fig. 13 displays The proportion of the number of times which a method obtained lower or equal RMSE than its competitors in a repeated holdout. Regression random machines is superior in at least 97% of the replications compared to any other considered method.

From these results, the regression random machines has a superior predictive performance to the extreme gradient boosting, SVM with linear kernel and deep learning neural networks models. On the other hand, the RRM showed a slightly superior general predictive performance, so the RRM can be a robust and competitive ensemble model comparable to light gradient boosting machine (LGBM).

Table 4
Table with the calculated values of RRMSE , MAE and COS over the test samples of the 30 repeated holdout from Simulated Scenarios 1–4.

n	Method	Model 1				Model 2			
		RMSE	MAE	COS	Time	RMSE	MAE	COS	Time
30	RRM	0.22	0.19	0.99	0.75	0.92	0.76	0.45	0.68
	SVR.Lap	0.20	0.16	0.99	2.95	0.94	0.80	0.38	3.90
	SVR.Lin	0.42	0.34	0.95	0.39	1.00	0.84	0.22	0.47
	SVR.Pol	0.12	0.09	1.00	0.34	0.92	0.75	0.46	0.43
	SVR.Gau	0.15	0.12	0.99	3.06	0.92	0.81	0.38	3.30
100	RRM	0.15	0.12	0.99	0.56	0.83	0.66	0.49	3.11
	SVR.Lap	0.15	0.12	0.99	3.11	0.91	0.72	0.23	6.05
	SVR.Lin	0.43	0.37	0.94	0.39	1.02	0.81	0.01	0.89
	SVR.Pol	0.11	0.09	1.00	0.48	0.87	0.68	0.35	3.62
	SVR.Gau	0.12	0.10	1.00	2.78	0.91	0.72	0.28	6.46
300	RRM	0.11	0.09	1.00	1.81	0.59	0.47	0.75	7.04
	SVR.Lap	0.12	0.10	1.00	8.89	0.65	0.52	0.70	6.84
	SVR.Lin	0.42	0.34	0.94	1.19	0.92	0.75	−0.05	0.72
	SVR.Pol	0.11	0.09	1.00	1.12	0.60	0.48	0.75	13.90
	SVR.Gau	0.11	0.09	1.00	7.40	0.65	0.52	0.69	4.96
1000	RRM	0.11	0.09	1.00	5.86	0.54	0.43	0.82	6.00
	SVR.Lap	0.11	0.09	1.00	31.40	0.59	0.47	0.78	13.30
	SVR.Lin	0.39	0.32	0.94	1.81	0.94	0.75	0.03	1.03
	SVR.Pol	0.11	0.09	1.00	2.70	0.54	0.44	0.82	11.90
	SVR.Gau	0.11	0.09	1.00	17.40	0.61	0.49	0.77	7.62

n	Method	Model 3				Model 4			
		RMSE	MAE	COS	Time	RMSE	MAE	COS	Time
30	RRM	0.85	0.69	0.88	0.54	2.57	2.20	0.93	0.58
	SVR.Lap	0.80	0.63	0.89	2.83	2.34	2.07	0.95	2.80
	SVR.Lin	0.79	0.63	0.90	0.43	2.22	1.79	0.94	0.56
	SVR.Pol	1.02	0.77	0.82	0.62	2.64	2.14	0.93	0.45
	SVR.Gau	0.85	0.67	0.88	2.81	2.53	2.20	0.93	2.71
100	RRM	0.61	0.48	0.93	1.12	2.09	1.72	0.95	1.18
	SVR.Lap	0.60	0.49	0.93	5.65	1.95	1.60	0.96	5.52
	SVR.Lin	0.62	0.48	0.92	0.72	2.21	1.84	0.94	1.00
	SVR.Pol	0.61	0.49	0.93	1.24	2.03	1.60	0.95	0.78
	SVR.Gau	0.58	0.46	0.93	6.15	2.06	1.69	0.95	5.55
300	RRM	0.58	0.47	0.93	1.19	1.82	1.50	0.97	1.22
	SVR.Lap	0.57	0.46	0.93	6.03	1.57	1.27	0.98	5.98
	SVR.Lin	0.65	0.52	0.91	0.59	2.08	1.71	0.96	0.61
	SVR.Pol	0.56	0.44	0.94	1.95	1.81	1.46	0.97	1.73
	SVR.Gau	0.57	0.46	0.93	4.42	1.59	1.26	0.98	4.92
1000	RRM	0.55	0.44	0.93	6.47	1.44	1.16	0.98	5.35
	SVR.Lap	0.55	0.44	0.93	29.80	1.02	0.80	0.99	28.50
	SVR.Lin	0.61	0.50	0.91	2.16	2.15	1.78	0.95	1.84
	SVR.Pol	0.55	0.44	0.93	7.42	1.77	1.50	0.97	6.61
	SVR.Gau	0.54	0.43	0.93	19.10	0.95	0.74	0.99	18.30

Table 5
Table with the calculated values of RMSE , MAE and COS over the test samples of the 30 repeated holdout from Simulated Scenarios 5–8.

n	Method	Model 5				Model 6			
		RMSE	MAE	COS	Time	RMSE	MAE	COS	Time
30	RRM	0.89	0.73	0.91	0.76	0.96	0.77	0.52	0.75
	SVR.Lap	0.80	0.69	0.93	3.36	0.69	0.55	0.74	3.01
	SVR.Lin	0.81	0.70	0.93	0.60	0.76	0.62	0.70	0.41
	SVR.Pol	0.80	0.66	0.92	0.52	0.82	0.65	0.63	0.40
	SVR.Gau	0.82	0.72	0.92	3.24	0.67	0.54	0.74	3.04
100	RRM	0.74	0.60	0.92	1.38	0.73	0.59	0.76	1.35
	SVR.Lap	0.71	0.58	0.92	5.97	0.71	0.58	0.77	5.81
	SVR.Lin	0.72	0.58	0.92	1.02	0.75	0.60	0.75	0.85
	SVR.Pol	0.85	0.70	0.89	3.35	0.72	0.59	0.76	2.15
	SVR.Gau	0.76	0.62	0.91	7.22	0.74	0.62	0.74	5.91
300	RRM	0.73	0.59	0.91	1.62	0.61	0.49	0.82	1.27
	SVR.Lap	0.71	0.57	0.91	6.76	0.59	0.48	0.83	6.15
	SVR.Lin	0.72	0.58	0.91	0.94	0.65	0.51	0.79	0.67
	SVR.Pol	0.79	0.62	0.89	38.40	0.59	0.48	0.83	5.65
	SVR.Gau	0.74	0.60	0.90	5.10	0.62	0.50	0.81	4.82
1000	RRM	0.68	0.55	0.92	2.66	0.56	0.45	0.85	3.61
	SVR.Lap	0.67	0.54	0.92	12.70	0.56	0.44	0.85	18.80
	SVR.Lin	0.71	0.57	0.91	1.10	0.61	0.48	0.82	1.51
	SVR.Pol	0.69	0.55	0.92	16.60	0.54	0.43	0.86	11.40
	SVR.Gau	0.69	0.55	0.92	7.64	0.56	0.45	0.85	11.70

n	Method	Model 7				Model 8			
		RMSE	MAE	COS	Time	RMSE	MAE	COS	Time
30	RRM	1.36	1.08	0.85	0.66	2.33	1.84	0.89	0.61
	SVR.Lap	1.17	0.96	0.88	2.80	1.76	1.44	0.94	3.04
	SVR.Lin	1.37	1.11	0.84	0.40	1.99	1.67	0.93	0.59
	SVR.Pol	0.93	0.76	0.93	0.41	2.01	1.59	0.91	0.45
	SVR.Gau	1.16	0.92	0.89	2.85	1.65	1.29	0.95	2.79
100	RRM	0.80	0.65	0.95	1.27	1.82	1.23	0.94	1.11
	SVR.Lap	0.75	0.61	0.96	5.98	1.73	1.20	0.94	5.69
	SVR.Lin	1.13	0.96	0.90	0.99	1.90	1.33	0.93	0.88
	SVR.Pol	0.58	0.47	0.98	1.93	1.88	1.31	0.93	2.08
	SVR.Gau	0.65	0.51	0.97	5.68	1.77	1.21	0.94	5.56
300	RRM	0.67	0.54	0.96	1.62	1.57	1.11	0.94	1.22
	SVR.Lap	0.62	0.50	0.96	5.81	1.43	1.03	0.95	6.05
	SVR.Lin	1.11	0.93	0.88	0.59	2.04	1.42	0.90	0.74
	SVR.Pol	0.55	0.44	0.97	2.06	1.53	1.15	0.94	5.20
	SVR.Gau	0.60	0.48	0.97	4.52	1.39	0.99	0.95	4.72
1000	RRM	0.58	0.46	0.97	4.36	1.18	0.84	0.98	2.96
	SVR.Lap	0.58	0.46	0.97	21.10	1.04	0.74	0.98	15.50
	SVR.Lin	1.07	0.88	0.88	1.55	1.87	1.41	0.94	1.22
	SVR.Pol	0.53	0.42	0.97	6.05	1.36	0.99	0.97	10.20
	SVR.Gau	0.56	0.44	0.97	13.40	1.06	0.78	0.98	9.67

In order to compare the predictive performance of the proposed method with other machine learning methods, exhaustive studies are necessary and are beyond the scope of this article. In this manner, Maia et al. (2021) show an exhaustive comparison between random machines and the random forest method.

5. Final comments

Random machines provide a new form to handle ensemble methods and support vector machines. The competitive predictive performance of the proposed method is proved theoretically and by several different experiments with artificial and real datasets. In general, regression random machines eliminate the problem of choosing a single appropriated kernel function and tuning its parameters in support vector regressions and is very competitive to other state-of-art machine learning methods.

Thus, the regression random machines combine different SVR models using different kernel functions and avoid the expensive computational cost of doing a grid or random search among the SVR models, besides reducing the general prediction error. To quantify this reduction, suppose many B models are calculated in a traditional bagging procedure and R as the number of kernels functions that will be evaluated and used in support vector models. In traditional bagging algorithms using SVR as base-models the number of total models that will be calculated to obtain the best results is given by $B \times K$ while using the regression random machines approach this number reduces to $B + K$. Using an example of $B = 100$ and $K = 400$, we have that the traditional bagging algorithm would take approximately four times the computational cost than the proposed Random Machines since the

Table 6
Comparison of computational time for considered models with different sample sizes.

n	Method	Time (in seconds)							
		Mod1	Mod2	Mod3	Mod4	Mod5	Mod6	Mod7	Mod8
30	RM	0.75	0.68	0.54	0.58	0.76	0.75	0.66	0.61
	SVR.Tuning	6.74	8.10	6.69	6.51	7.72	6.86	6.46	6.87
100	RM	0.56	3.11	1.12	1.18	1.38	1.35	1.27	1.11
	SVR.Tuning	6.76	17.02	13.76	12.85	17.56	14.72	14.58	14.21
300	RM	1.81	7.04	1.19	1.22	1.62	1.27	1.62	1.22
	SVR.Tuning	18.60	26.42	12.99	13.24	51.20	17.29	12.98	16.71
1000	RM	5.86	6.00	6.47	5.35	2.66	3.61	4.36	2.96
	SVR.Tuning	53.31	33.85	58.48	55.25	38.04	43.41	42.10	36.59

Table 7
Description of the twenty six regression datasets.

ID	Data Set	# Instances	# Features	# Cont. : # Categ.	ID	Data Set	# Instances	# Features	# Cont. : # Categ.
1	*abalone*	4177	7	6:1	14	*machines*	208	7	6:1
2	*airbnb*	10498	17	17:0	15	*mpg*	398	6	3:3
3	*airfoil*	1502	5	6:0	16	*ozone*	330	8	8:0
4	*boston housing*	505	13	12:1	17	*parkinson*	1040	26	25:1
5	*cars*	50	1	1:0	18	*petrol*	31	4	4:0
6	*cement*	12	4	4:0	19	*pyrim*	74	27	27:0
7	*concrete*	1030	8	8:0	20	*servo*	167	4	2:2
8	*cpus*	208	6	6:0	21	*slump*	102	7	7:0
9	*friedman#1*	500	10	10:0	22	*space ga*	3107	6	6:0
10	*friedman#2*	500	4	4:0	23	*stormer*	22	2	2:0
11	*friedman#3*	500	4	4:0	24	*taiwan*	414	6	6:0
12	*geysers*	298	1	1:0	25	*triazine*	185	60	60:0
13	*hills*	34	2	2:0	26	*yatch*	308	6	6:0

Table 8
Predictive performance metrics of the considered methods to the regression task in the Bolsa familia dataset.

Metric	RRM	DP-MLP$_1$	DP-MLP$_2$	LGBM	XGBM	SVM$_{LIN}$	SVM$_{POL}$	SVM$_{GAU}$
RMSE	**0.280**	0.292	0.291	0.287	0.302	0.300	0.288	0.288
MAE	**0.209**	0.219	0.220	0.214	0.226	0.224	0.214	0.214
COS	**0.960**	0.957	0.957	0.958	0.954	0.954	0.958	0.958

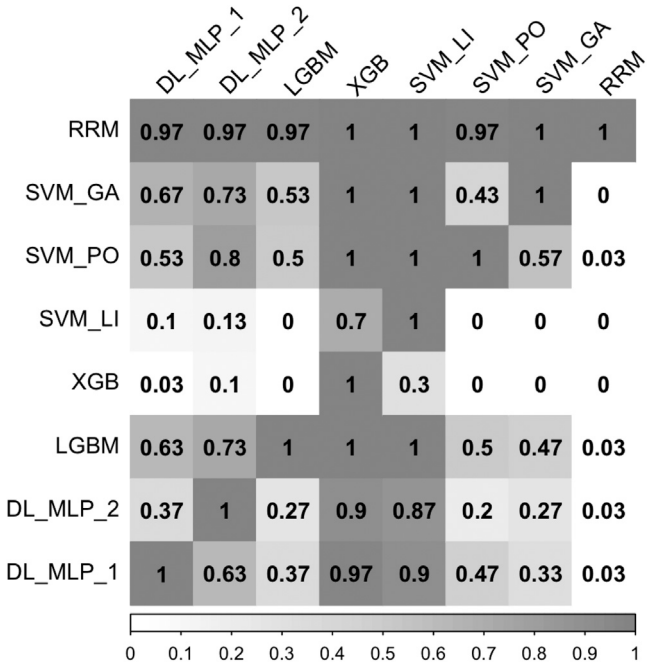

Fig. 13. The proportion of the number of times which a method obtained lower RMSE than its competitors in a repeated holdout.

ratio of calculated models is 400/104 (i.e: four times faster). The reduction of the computational time using RRM compared to tuning processes of the SVR is very expressive specially when training sample size increases. Furthermore, the results from RRM explored the strength and correlation characteristics in the bagging procedure, obtaining simultaneously lower generalization error and agreement, instead of traditional ensemble procedures using SVR as base models that cannot achieve them at the same time.

Additionally, this methodology can be explored in other contexts and can be applied to any practical statistical learning regression problem. Future theoretical studies may be done concerning the use of other and more kernel functions in the bagging step, besides other weighting function approaches as well as extensions for large datasets.

Declaration of competing interest

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

Acknowledgments

The authors gratefully acknowledge the financial support of the Brazilian research funding agencies CAPES (Federal Agency for the Support and Improvement of Higher Education). The authors are also thankful to the unknown reviewers that reviewed and suggested changes which significantly improved the final version of this manuscript. M.M.'s work was supported by a Science Foundation Ireland Career Development Award grant 17/CDA/4695.

References

Abuella, M., & Chowdhury, B. (2017). Random forest ensemble of support vector regression models for solar power forecasting. In *2017 IEEE power & energy society innovative smart grid technologies conference* (pp. 1–5). IEEE.

Alzubi, J. A. (2015). Diversity based improved bagging algorithm. In *Proceedings of the the international conference on engineering & MIS 2015* (pp. 1–5).

Amarnath, B., Balamurugan, S., & Alias, A. (2016). Review on feature selection techniques and its impact for effective data classification using UCI machine learning repository dataset. *Journal of Engineering Science and Technology*, *11*(11), 1639–1646.

Ara, A., Maia, M., Louzada, F., & Macêdo, S. (2021a). Random machines: A bagged-weighted support vector model with free kernel choice. *Journal of Data Science*, [ISSN: 1680-743X] 1–20. http://dx.doi.org/10.6339/21-JDS1014.

Araque, O., Corcuera-Platas, I., Sánchez-Rada, J. F., & Iglesias, C. A. (2017). Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications*, *77*, 236–246.

Ayat, N.-E., Cheriet, M., & Suen, C. Y. (2005). Automatic model selection for the optimization of SVM kernels. *Pattern Recognition*, *38*(10), 1733–1745.

Bergstra, J. S., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems* (pp. 2546–2554).

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*(Feb), 281–305.

Bhatnagar, V., Bhardwaj, M., Sharma, S., & Haroon, S. (2014). Accuracy–diversity based pruning of classifier ensembles. *Progress in Artificial Intelligence*, *2*(2–3), 97–111.

Bi, Y. (2012). The impact of diversity on the accuracy of evidential classifier ensembles. *International Journal of Approximate Reasoning*, *53*(4), 584–607.

Borra, S., & Di Ciaccio, A. (2002). Improving nonparametric regression methods by bagging and boosting. *Computational Statistics & Data Analysis*, *38*(4), 407–420.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on computational learning theory* (pp. 144–152). ACM.

Brahim, A. B., & Limam, M. (2018). Ensemble feature selection for high dimensional data: a new method and a comparative study. *Advances in Data Analysis and Classification*, *12*(4), 937–952.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Breiman, L., et al. (1996). Heuristics of instability and stabilization in model selection. *The Annals of Statistics*, *24*(6), 2350–2383.

Brown, G., Yao, X., Wyatt, J., Wersing, H., & Sendhoff, B. (2002). Exploiting ensemble diversity for automatic feature extraction. In *Proceedings of the 9th international conference on neural information processing, Vol. 4* (pp. 1786–1790). IEEE.

Cavalcanti, G. D., Oliveira, L. S., Moura, T. J., & Carvalho, G. V. (2016). Combining diversity measures for ensemble pruning. *Pattern Recognition Letters*, *74*, 38–45.

Chandra, A., & Yao, X. (2006). Evolving hybrid ensembles of learning machines for better generalisation. *Neurocomputing*, *69*(7–9), 686–700.

Chapelle, O., & Vapnik, V. (2000). Model selection for support vector machines. In *Advances in neural information processing systems* (pp. 230–236).

Chen, L., Tran, H., Batra, R., Kim, C., & Ramprasad, R. (2019). Machine learning models for the lattice thermal conductivity prediction of inorganic materials. *Computational Materials Science*, *170*, Article 109155.

Cherkassky, V., & Ma, Y. (2004). Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks*, *17*(1), 113–126.

Chollet, F., Allaire, J., et al. (2017). R interface to Keras. GitHub, https://github.com/rstudio/keras.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Courant, R., & Hilbert, D. (1953). *Methods of mathematical physics, Vol. I* (p. 106). New York: Interscience Publ Inc..

Croux, C., Joossens, K., & Lemmens, A. (2007). Trimmed bagging. *Computational Statistics & Data Analysis*, *52*(1), 362–368.

Cunningham, P., & Carney, J. (2000). Diversity versus quality in classification ensembles based on feature selection. In *European conference on machine learning* (pp. 109–116). Springer.

Delbari, M., Sharifazari, S., & Mohammadi, E. (2019). Modeling daily soil temperature over diverse climate conditions in Iran—a comparison of multiple linear regression and support vector regression techniques. *Theoretical and Applied Climatology*, *135*(3–4), 991–1001.

Deng, Y.-F., Jin, X., & Zhong, Y.-X. (2005). Ensemble SVR for prediction of time series. In *2005 international conference on machine learning and cybernetics, Vol. 6* (pp. 3528–3534). IEEE.

Devi, B., Kumar, S., Shankar, V. G., et al. (2019). AnaData: A novel approach for data analytics using random forest tree and SVM. In *Computing, communication and signal processing* (pp. 511–521). Springer.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A. J., & Vapnik, V. (1997). Support vector regression machines. In *Advances in neural information processing systems* (pp. 155–161).

Dua, D., & Graff, C. (2017). UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences, URL http://archive.ics.uci.edu/ml.

Duan, K., Keerthi, S. S., & Poo, A. N. (2003). Evaluation of simple performance measures for tuning SVM hyperparameters. *Neurocomputing*, *51*, 41–59.

Efron, B., & Tibshirani, R. (1986). Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical Science*, 54–75.

Evgeniou, T., Pontil, M., & Elisseeff, A. (2004). Leave one out error, stability, and generalization of voting combinations of classifiers. *Machine Learning*, *55*(1), 71–97.

Féret, J.-B., Le Maire, G., Jay, S., Berveiller, D., Bendoula, R., Hmimina, G., et al. (2019). Estimating leaf mass per area and equivalent water thickness based on leaf optical properties: Potential and limitations of physical modeling and machine learning. *Remote Sensing of Environment*, *231*, Article 110959.

Freund, Y., Schapire, R., & Abe, N. (1999). A short introduction to boosting. *Journal-Japanese Society for Artificial Intelligence*, *14*(771–780), 1612.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *The Annals of Statistics*, 1189–1232.

Friedrichs, F., & Igel, C. (2005). Evolutionary tuning of multiple SVM parameters. *Neurocomputing*, *64*, 107–117.

Gul, A., Perperoglou, A., Khan, Z., Mahmoud, O., Miftahuddin, M., Adler, W., et al. (2018). Ensemble of a subset of kNN classifiers. *Advances in Data Analysis and Classification*, *12*(4), 827–840.

Heinermann, J., & Kramer, O. (2014). Precise wind power prediction with SVM ensemble regression. In *International conference on artificial neural networks* (pp. 797–804). Springer.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, *29*(6), 82–97.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(8), 1–22.

Huang, M.-W., Chen, C.-W., Lin, W.-C., Ke, S.-W., & Tsai, C.-F. (2017). SVM and SVM ensembles in breast cancer prediction. *PLoS One*, *12*(1), Article e0161501.

Jebara, T. (2004). Multi-task feature and kernel selection for SVMs. In *Proceedings of the twenty-first international conference on machine learning* (p. 55). ACM.

Karatzoglou, A., Smola, A., Hornik, K., & Zeileis, A. (2004). Kernlab – An S4 package for kernel methods in R. *Journal of Statistical Software*, *11*(9), 1–20, URL http://www.jstatsoft.org/v11/i09/.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., et al. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, *30*.

Khan, M. M. R., Arif, R. B., Siddique, M. A. B., & Oishe, M. R. (2018). Study and observation of the variation of accuracies of KNN, SVM, LMNN, ENN algorithms on eleven different datasets from UCI machine learning repository. In *2018 4th international conference on electrical engineering and information & communication technology* (pp. 124–129). IEEE.

Khan, Z., Gul, A., Perperoglou, A., Miftahuddin, M., Mahmoud, O., Adler, W., et al. (2019). Ensemble of optimal trees, random forest and random projection ensemble classification. *Advances in Data Analysis and Classification*, 1–20.

Khosravi, A., Koury, R., Machado, L., & Pabon, J. (2018). Prediction of wind speed and wind direction using artificial neural network, support vector regression and adaptive neuro-fuzzy inference system. *Sustainable Energy Technologies and Assessments*, *25*, 146–160.

Kim, H.-C., Pang, S., Je, H.-M., Kim, D., & Bang, S.-Y. (2002). Support vector machine ensemble with bagging. In *International workshop on support vector machines* (pp. 397–408). Springer.

Klambauer, G., Unterthiner, T., Mayr, A., & Hochreiter, S. (2017). Self-normalizing neural networks. *Advances in Neural Information Processing Systems*, *30*.

Kuncheva, L. I. (2003). That elusive diversity in classifier ensembles. In *Iberian conference on pattern recognition and image analysis* (pp. 1126–1138). Springer.

Lam, L. (2000). Classifier combinations: implementations and theoretical issues. In *International workshop on multiple classifier systems* (pp. 77–86). Springer.

Lee, S., Ahmad, A., & Jeon, G. (2018). Combining bootstrap aggregation with support vector regression for small blood pressure measurement. *Journal of Medical Systems*, *42*(4), 63.

Li, Y., Che, J., & Yang, Y. (2018). Subsampled support vector regression ensemble for short term electric load forecasting. *Energy*, *164*, 160–170.

Maia, M., Azevedo, A. R., & Ara, A. (2021). Predictive comparison between random machines and random forests. *Journal of Data Science*, *19*(4), 593–614.

Mazaheri, T., Sun, B., Scher-Zagier, J., Thind, A., Magee, D., Ronhovde, P., et al. (2019). Stochastic replica voting machine prediction of stable cubic and double perovskite materials and binary alloys. *Physical Review Materials*, *3*(6), Article 063802.

McNally, S., Roche, J., & Caton, S. (2018). Predicting the price of bitcoin using machine learning. In *2018 26th euromicro international conference on parallel, distributed and network-based processing* (pp. 339–343). IEEE.

Meier, L., Van de Geer, S., Bühlmann, P., et al. (2009). High-dimensional additive modeling. *The Annals of Statistics*, *37*(6B), 3779–3821.

Mendes-Moreira, J., Soares, C., Jorge, A. M., & Sousa, J. F. D. (2012). Ensemble approaches for regression: A survey. *Acm Computing Surveys (Csur)*, *45*(1), 10.

Okujeni, A., van der Linden, S., Suess, S., & Hostert, P. (2016). Ensemble learning from synthetically mixed training data for quantifying urban land cover with support vector regression. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *10*(4), 1640–1650.

Park, C., Koo, J.-Y., Kim, S., Sohn, I., & Lee, J. W. (2008). Classification of gene functions using support vector machine for time-course gene expression data. *Computational Statistics & Data Analysis*, *52*(5), 2578–2587.

Paz, H., Maia, M., Moraes, F., Lustosa, R., Costa, L., Macêdo, S., et al. (2020). Local processing of massive databases with R: A national analysis of a Brazilian social programme. *Stats*, *3*(4), 444–464.

Rakesh, K., & Suganthan, P. N. (2017). An ensemble of kernel ridge regression for multi-class classification. *Procedia Computer Science*, *108*, 375–383.

Ripley, B., Venables, B., Bates, D. M., Hornik, K., Gebhardt, A., Firth, D., et al. (2013). Package 'mass'. *Cran R*, *538*, 113–120.

Roy, M.-H., & Larocque, D. (2012). Robustness of random forests for regression. *Journal of Nonparametric Statistics*, *24*(4), 993–1006.

Scornet, E. (2016). Random forests and kernel methods. *IEEE Transactions on Information Theory*, *62*(3), 1485–1500.

Shivaswamy, P. K., Chu, W., & Jansche, M. (2007). A support vector approach to censored targets. In *Seventh IEEE international conference on data mining* (pp. 655–660). IEEE.

Sluban, B., & Lavrač, N. (2015). Relating ensemble diversity and performance: A study in class noise detection. *Neurocomputing*, *160*, 120–131.

Smutz, C., & Stavrou, A. (2016). When a tree falls: Using diversity in ensemble classifiers to identify evasion in malware detectors. In *NDSS*.

Song, M., Breneman, C. M., Bi, J., Sukumar, N., Bennett, K. P., Cramer, S., et al. (2002). Prediction of protein retention times in anion-exchange chromatography systems using support vector regression. *Journal of Chemical Information and Computer Sciences*, *42*(6), 1347–1357.

Stork, J., Ramos, R., Koch, P., & Konen, W. (2015). SVM ensembles are better when different kernel types are combined. In *Data science, learning by latent structures, and knowledge discovery* (pp. 191–201). Springer.

Travis-Lumer, Y., & Goldberg, Y. (2021). Kernel machines for current status data. *Machine Learning*, *110*(2), 349–391.

Turney, P. (1995). Bias and the quantification of stability. *Machine Learning*, *20*(1–2), 23–33.

Valentini, G., & Dietterich, T. G. (2004). Bias-variance analysis of support vector machines for the development of SVM-based ensemble methods. *Journal of Machine Learning Research*, *5*(Jul), 725–775.

Van der Laan, M. J., Polley, E. C., & Hubbard, A. E. (2007). Super learner. *Statistical Applications in Genetics and Molecular Biology*, *6*(1).

Van Wezel, M., & Potharst, R. (2007). Improved customer choice predictions using ensemble methods. *European Journal of Operational Research*, *181*(1), 436–452.

Vapnik, V., & Chervonenkis, A. (1974). *Theory of pattern recognition*. Nauka, Moscow.

Wang, H., Zheng, B., Yoon, S. W., & Ko, H. S. (2018). A support vector machine-based ensemble algorithm for breast cancer diagnosis. *European Journal of Operational Research*, *267*(2), 687–699.

Wu, C.-H., Ho, J.-M., & Lee, D.-T. (2004). Travel-time prediction with support vector regression. *IEEE Transactions on Intelligent Transportation Systems*, *5*(4), 276–281.

Wu, Z., Li, N., Peng, J., Cui, H., Liu, P., Li, H., et al. (2018). Using an ensemble machine learning methodology-bagging to predict occupants' thermal comfort in buildings. *Energy and Buildings*, *173*, 117–127.

Wu, C.-H., Tzeng, G.-H., & Lin, R.-H. (2009). A novel hybrid genetic algorithm for kernel function and parameter optimization in support vector regression. *Expert Systems with Applications*, *36*(3), 4725–4735.

Xiao, X., Zhang, T., Zhong, X., Shao, W., & Li, X. (2018). Support vector regression snow-depth retrieval algorithm using passive microwave remote sensing data. *Remote Sensing of Environment*, *210*, 48–64.

Ye, R., & Suganthan, P. N. (2012). A kernel-ensemble bagging support vector machine. In *2012 12th international conference on intelligent systems design and applications* (pp. 847–852). IEEE.

Yildizer, E., Balci, A. M., Hassan, M., & Alhajj, R. (2012). Efficient content-based image retrieval using multiple support vector machines ensemble. *Expert Systems with Applications*, *39*(3), 2385–2396.