scientific reports

OPEN

Boosting ridge for the extreme learning machine globally optimised for classification and regression problems

Carlos Peralez-González, Javier Pérez-Rodríguez✉ & Antonio M. Durán-Rosal

This paper explores the boosting ridge (BR) framework in the extreme learning machine (ELM) community and presents a novel model that trains the base learners as a global ensemble. In the context of Extreme Learning Machine single-hidden-layer networks, the nodes in the hidden layer are preconfigured before training, and the optimisation is performed on the weights in the output layer. The previous implementation of the BR ensemble with ELM (BRELM) as base learners fix the nodes in the hidden layer for all the ELMs. The ensemble learning method generates different output layer coefficients by reducing the residual error of the ensemble sequentially as more base learners are added to the ensemble. As in other ensemble methodologies, base learners are selected until fulfilling ensemble criteria such as size or performance. This paper proposes a global learning method in the BR framework, where base learners are not added step by step, but all are calculated in a single step looking for ensemble performance. This method considers (i) the configurations of the hidden layer are different for each base learner, (ii) the base learners are optimised all at once, not sequentially, thus avoiding saturation, and (iii) the ensemble methodology does not have the disadvantage of working with strong classifiers. Various regression and classification benchmark datasets have been selected to compare this method with the original BRELM implementation and other state-of-the-art algorithms. Particularly, 71 datasets for classification and 52 for regression, have been considered using different metrics and analysing different characteristics of the datasets, such as the size, the number of classes or the imbalanced nature of them. Statistical tests indicate the superiority of the proposed method in both regression and classification problems in all experimental scenarios.

In the last decade, Extreme Learning Machine (ELM)[1] has become a popular methodology in Machine Learning challenging problems, for instance, brain-computer interfaces[2], the prediction of the remaining rolling bearing useful life[3], the origin detection of fennel which is of great importance in food flavouring[4], the COVID-19-pneumonia prediction[5], EGG classification for brain-computer interface[6], water network management[7], and wheat yield prediction[8], among others. ELM theories claim that the hidden layer parameters, that is, the weight and bias in single-hidden layer feed-forward networks, do not need to be tuned, but they can be generated randomly, independently of the training dataset[9]. Thus, only the output weights are computed in a single step by employing the least-squares estimated solution. Due to this random initialisation, ELM training speed is more efficient compared to the traditional solvers for neural networks, for instance, those based on back-propagation[10,11], without losing performance, and even improving it.

One of the drawbacks of ELM models is that it requires a high number of neurons for the hidden layer because the nonlinear combination of features is explored randomly[12]. Due to this, several methods have been investigated for reducing this randomness without increasing the computation time or the algorithm's complexity, such as pruning[13], swarm optimisation[14,15] and ensemble learning methods.

In this context, several ensemble methods for ELM models have been proposed, e.g., ensembles for regression[16], fuzzy ensembles for big data classification[17], deep ensembles for time series forecasting[18], incremental Meta-ELM with error feedback[19] or weighted kernel ELM ensembles for imbalanced datasets[20]. Furthermore, many ELM ensemble methods have been applied to real-world problems, such as the prediction of ocean wave height[21], human activity recognition[22], calibration of near-infrared spectroscopy[23] or birdsong recognition[24]. In general, ensembles aim to improve the generalisation error using a mixture of classifiers or regressors, known as

Department of Quantitative Methods, Universidad Loyola Andalucía, Córdoba, Spain. ✉email: jperez@uloyola.es

base learners in the ensemble learning framework. The performance improvement is associated with diversity among the base predictors, i.e. it is essential for the generalisation of the ensemble that the base learners disagree as much as possible[25]. There are many ways to combine individual predictions. Thus several voting methods have been proposed to improve the efficiency of these ensembles, such as Bagging[26], Boosting[27], incremental learning system using local linear experts[28] or a variation of Boosting constructed from a functional gradient descent algorithm with the L2-loss function[29], among others. The ensemble methodologies known as Bagging and Boosting are the most widely used approaches, mainly because of their ease of application and their ensemble performance[30]. The key to these ensemble methodologies lies in the training data to generate diversity. In this way, diverse solutions to the optimisation problem associated with the base predictors are implicitly sought through data sampling[31].

Specifically, in the field of Boosting philosophy, a particularly interesting algorithm is Boosting Ridge (BR)[32]. This ensemble algorithm, designed originally for regression problems, trains the base learners sequentially, setting the residual of the previous predictor as the training target. The first base learner is the predictor for the original target. Subsequently, the error between the prediction on the training set and the target is calculated, and this residual is the new target. The second predictor is trained with this residual. After calculating the error between the second predictor and the first residual, a third residual is calculated, which is the target of the next predictor. The process is repeated until the number of base learners is reached. BR shows its importance in many applications, such as early-stage breast cancer detection[33], microarray survival models[34] and criminal recidivism predictions[35].

The addition of base learners does not continuously improve the ensemble since there is a trade-off between diversity among the base learners and the final ensemble performance[36]. Furthermore, in the boosting methodology, although each base learner is added to reduce the error of the previous ones, the saturation of the base learners eventually appears. The saturation occurs when the ensemble cannot improve the generalisation error despite introducing more and more base learners. Also, if the number of base learners is fixed, saturation or even overfitting could be produced because the base learners become stronger (more accurate). It is given that increasing the number of hidden neurons reduces the diversity in the ensemble[37], which is needed to improve the ensemble performance[25]. To overcome the saturation and to give an approach model selection,[38] proposes the use of genetic algorithms to select the optimal number of base learners involved in the ensemble,[39] proposes an adaptative stopping rule via adjusting the regularisation parameter, and[40] relies on diversity measures to establish the upper bound of a number of base learners.

Like other ensemble methodologies[36,41], BR aims to train each base predictor separately and then combine their results. BR algorithm for ELM-based learners (BRELM) was initially proposed by Ran et al.[42]. With this background and to overcome the main drawbacks mentioned above, this paper proposes a new boosting algorithm that removes the need to add base learners sequentially, leading to saturation. Also, using strong instead of weak base classifiers does not worsen the ensemble's performance. For this, several predictors are optimised at once to calculate the optimised ensemble parameters globally. The formulation of the error function allows the development of an analytical solution for the parameters of the ELM-based learners to find the weights of the output layers for each base learner in a single step. Moreover, this ensemble learning method achieves better results than the sequential BR, as the error is optimised globally in the ensemble and not for each base learner.

Summarising, the novel contributions of this work are:

- The optimisation of the weights of the output layer of a Boosting Ridge for Extreme Learning Machine ensemble in a single step instead of iteratively, with the objective of reducing the generalisation error.
- The use of different input layers mappings with different parameters for their hidden layers, made possible by the new optimisation approach resulting in the so-called Generalised Global BRELM (GGBRELM), tends to a better diversity of the ensemble.
- Avoid the problem of ensemble saturation and overtraining by making the new proposal work well when the base classifiers become stronger. For example, it is known that by increasing the number of neurons in the ELM networks of the base learners, each one achieves good performance, but, in return, the ensemble's performance is reduced. With the new proposal, this problem is solved.
- The application of the methodology to more than 120 classification and regression datasets from different domains shows that the proposal works better than the state-of-the-art methods and can be applied to any real-world problem.
- The performance of the proposed methodology analysis considers different dataset properties such as size, number of classes or imbalance.

This paper is organised as follows: "State-of-the-art algorithms" Section summarises the notation and formulation of the ELM, BR and BRELM algorithms. "Methodology of the proposal" Section develops the proposed methodology about the globalisation of BRELM and its generalised version GGBRELM, shows a graphical comparison of the methodologies, and includes an analysis of their computational costs. The experimental design is set in Section experimental design, while "Discussion of the results" Section explains the most highlighted results, including statistical analysis. Finally, "Conclusions" Section collects the main conclusions obtained in the work.

State-of-the-art algorithms

This section introduces the notation and formulation of the two algorithms on which this proposal is based, i.e., ELM predictor and BR ensemble methodology.

Extreme learning machine. For a simple supervised learning problem, dataset $\mathscr{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{N}$ consists in a set of N patterns, each one with a vector of features, $\mathbf{x}_n$ and target associated, $\mathbf{y}_n$.

- $\mathbf{x}_n \in \mathbb{R}^K$ is the data information for the n-th pattern, where K is the number of input variables.
- $\mathbf{y}_n$ is the target variable for the n-th pattern. In case of regression problems, $y_n \in \mathbb{R}$ since it is a number. In classification problems with J classes, the target can be expressed as "1-of-J" encoding, $\mathbf{y}_n \in \mathbb{R}^J$. Each component j of $\mathbf{y}_n$ is $y_{j,n} = 1$ if n-th pattern belongs to class j and $y_{j,n} = 0$ otherwise.

Using "1-of-J" encoding, a classification can be rewritten as a multi-regression problem. Thus, ELM model is explained for regression problems in this subsection, and the explanation for classification is summed up at the end. A predictor $f : \mathbb{R}^K \rightarrow \mathbb{R}$ inferring a function that maps an input n-th pattern $\mathbf{x}_n$ to an output target y_n, using relationships from labeled dataset $\mathscr{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$. In particular, Extreme Learning Machine (ELM) model build this function:

$$f(\mathbf{x}) = \mathbf{h}'(\mathbf{x})\boldsymbol{\beta}, \tag{1}$$

where:

- $\mathbf{h} : \mathbb{R}^K \rightarrow \mathbb{R}^D$ is a non-linear mapping of the input layer. It transforms the pattern $\mathbf{x}_n$ from the original feature space $\mathbb{R}^K$ to the transformed space $\mathbb{R}^D$, where D is the number of neurons in the hidden layer. This mapping is explicitly computed as
$$\mathbf{h}(\mathbf{x}) = (\phi_d(\mathbf{x}; \mathbf{w}_d, b_d),\ d = 1, \ldots, D), \tag{2}$$
with $\phi : \mathbb{R}^K \rightarrow \mathbb{R}$ as the activation function for the neuron d, and the weights $\mathbf{w}_d$ and biases b_d are randomly generated.
- $\boldsymbol{\beta} : \mathbb{R}^D$ is the vector of weights in the output layer, that are found in the optimisation problem:
$$\min_{\boldsymbol{\beta} \in \mathbb{R}^D} \ \|\mathbf{H}\boldsymbol{\beta} - \mathbf{Y}\|^2 + C\|\boldsymbol{\beta}\|^2, \tag{3}$$
where $\mathbf{H} = \left(\mathbf{h}'(\mathbf{x}_1), \ldots, \mathbf{h}'(\mathbf{x}_N)\right) \in \mathbb{R}^{N \times D}$ is the output of the hidden layer for the training patterns, $\mathbf{Y} = \begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_N \end{pmatrix} \in \mathbb{R}^N$ is the matrix with the desired targets and $C > 0$ is an user-specified term, that controls the regularisation in the model[12].

Equation (3) represents a convex minimisation problem with error and regularisation terms. The error term $\|\mathbf{H}\boldsymbol{\beta} - \mathbf{Y}\|^2$ adjusts the coefficient vector $\boldsymbol{\beta}$ in order to minimise the error of the prediction $\mathbf{Y}$, while the regularisation term $\|\boldsymbol{\beta}_j\|^2$ is included to avoid over-fitting in the model[43].

The optimal solution for the model is the minimum of the convex objective function in Eq. (3), and it is obtained by deriving and equaling to 0:

$$\boldsymbol{\beta} = \left(\mathbf{H}'\mathbf{H} + C\mathbf{I}\right)^{-1}\mathbf{H}'\mathbf{Y}. \tag{4}$$

For a classification problem, there are J minimisation problems as Eq. (3). The predicted class corresponds to the vector component with the highest value, that is

$$\arg \max_{j=1,\ldots,J} (\mathbf{h}(\mathbf{x})\boldsymbol{\beta}_1, \ldots, \mathbf{h}(\mathbf{x})\boldsymbol{\beta}_j, \ldots, \mathbf{h}(\mathbf{x})\boldsymbol{\beta}_J) \tag{5}$$

Boosting ridge regression (linear model). From a linear regression model,

$$f(\mathbf{x}) = \mathbf{x}'\boldsymbol{\beta}, \tag{6}$$

and its associated minimisation problem

$$\min_{\boldsymbol{\beta}} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{Y}\|^2 + C\|\boldsymbol{\beta}\|^2, \tag{7}$$

with $\mathbf{X} = \begin{pmatrix} \mathbf{x}_1' \\ \vdots \\ \mathbf{x}_N' \end{pmatrix} \in \mathbb{R}^{N \times K}$, Tutz et al.[32] proposed BR Regression as ensemble learning method that reduces sequentially the residual of the ensemble prediction,

$$
\begin{aligned}
&\min_{\boldsymbol{\beta}^{(1)}} \|\mathbf{X}\boldsymbol{\beta}^{(1)} - \mathbf{Y}\|^2 + C\|\boldsymbol{\beta}^{(1)}\|^2; \qquad \mu_1 = \mathbf{Y} - \mathbf{X}\boldsymbol{\beta}^{(1)},\\
&\min_{\boldsymbol{\beta}^{(2)}} \|\mathbf{X}\boldsymbol{\beta}^{(2)} - \mu_1\|^2 + C\|\boldsymbol{\beta}^{(2)}\|^2; \qquad \mu_2 = \mu_1 - \mathbf{X}\boldsymbol{\beta}^{(2)},\\
&\min_{\boldsymbol{\beta}^{(3)}} \|\mathbf{X}\boldsymbol{\beta}^{(3)} - \mu_2\|^2 + C\|\boldsymbol{\beta}^{(3)}\|^2; \qquad \mu_3 = \mu_2 - \mathbf{X}\boldsymbol{\beta}^{(3)},\\
&\qquad\qquad\qquad\qquad\vdots
\end{aligned} \tag{8}
$$

For an ensemble with S base learners, the prediction of the BR Regression model is

$$
f(\mathbf{x}) = \sum_{s=1}^{S} \mathbf{x}'\boldsymbol{\beta}^{(s)} = \mathbf{x}' \sum_{s=1}^{S} \boldsymbol{\beta}^{(s)}. \tag{9}
$$

Boosting ridge extreme learning machine.

BR ensemble learning methodology was adapted to the ELM community by Ran et al.[42]. The prediction of this sequential ensemble, BRELM, of S base learners is the following linear combination:

$$
f(\mathbf{x}) = \sum_{s=1}^{S} f^{(s)}(\mathbf{x}) = \mathbf{h}'(\mathbf{x}) \sum_{s=1}^{S} \boldsymbol{\beta}^{(s)}. \tag{10}
$$

The first base learner $s = 1$ is the standard ELM solution from Eq. (3). Later, the s-th base learner training stage uses all the data, but the target $\mu^{(s)}$ is the residual of the previous base learner predictions,

$$
\mu^{(s)} = \mathbf{Y} - \sum_{s'=1}^{s-1} \mathbf{H}\boldsymbol{\beta}^{(s')}. \tag{11}
$$

Therefore, the minimisation problem of the s-th base learner is

$$
\min_{\boldsymbol{\beta}^{(s)}} \|\mathbf{H}\boldsymbol{\beta}^{(s)} - \mu^{(s)}\|^2 + C\|\boldsymbol{\beta}^{(s)}\|^2, \tag{12}
$$

and the solution for the output layer of the s-th base learner is

$$
\boldsymbol{\beta}^{(s)} = (\mathbf{H}'\mathbf{H} + C\mathbf{I})^{-1}\mathbf{H}'\left(\mathbf{Y} - \mathbf{H}\sum_{s'=1}^{s-1} \boldsymbol{\beta}^{(s')}\right). \tag{13}
$$

Methodology of the proposal

In this section, the Globalisation of the BRELM is proposed, along with an enhanced version called Generalised Global BRELM (GGBRELM). A methodological graphical comparison is also included. And finally, a theoretical analysis of the methodologies' computational complexities is discussed.

The main hypothesis of this work is that the methodology based on the optimisation of all the base learners in a single step will improve the generalisation error of the ensemble. Thus, considering that this procedure will avoid the saturation of the ensemble, and therefore, for a high number of neurons (strong ELM base learners), the ensemble performance will not be reduced. Besides, the use of different input layer weights and, therefore, different mapping functions ($\mathbf{h}^{(s)}$) between the different base predictors will lead to more diversity in the ensemble.

Global boosting ridge for extreme learning machine.

The main idea behind BRELM is to reduce sequentially the error produced by the ensemble. This proposal, Global BRELM, presents the problem for each s-th base learner as the error reduction of the other base learners of the ensemble.

$$
f(\mathbf{x}) = \sum_{s=1}^{S} f^{(s)}(\mathbf{x}), \tag{14}
$$

$$
\min_{\boldsymbol{\beta}^{(s)}} \|\mathbf{H}\boldsymbol{\beta}^{(s)} - (\mathbf{Y} - \sum_{s' \neq s}^{S} \mathbf{H}\boldsymbol{\beta}^{(s')})\|^2 + C\|\boldsymbol{\beta}^{(s)}\|^2. \tag{15}
$$

Deriving respect with $\boldsymbol{\beta}^{(s)}$ and equal to 0, some terms depend on $\boldsymbol{\beta}^{(s)}$ while other ones depend on $\boldsymbol{\beta}^{(s')}$, $s' = 1, \ldots, S, s' \neq s$,

$$\left(\mathbf{H}'\mathbf{H}+C\mathbf{I}\right)\boldsymbol{\beta}^{(s)}+\mathbf{H}'\sum_{s'\neq s}^{S}\mathbf{H}\boldsymbol{\beta}^{(s')}-\mathbf{H}'\mathbf{Y}=0. \tag{16}$$

From the previous equation, a system of equations can be set up,

$$\begin{pmatrix} \mathbf{H}'\mathbf{H}+C\mathbf{I} & \mathbf{H}'\mathbf{H} & \dots & \mathbf{H}'\mathbf{H} \\ \mathbf{H}'\mathbf{H} & \mathbf{H}'\mathbf{H}+C\mathbf{I} & \dots & \mathbf{H}'\mathbf{H} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{H}'\mathbf{H} & \mathbf{H}'\mathbf{H} & \dots & \mathbf{H}'\mathbf{H}+C\mathbf{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{\beta}^{(1)} \\ \boldsymbol{\beta}^{(2)} \\ \vdots \\ \boldsymbol{\beta}^{(s)} \end{pmatrix} = \begin{pmatrix} \mathbf{H}'\mathbf{Y} \\ \mathbf{H}'\mathbf{Y} \\ \vdots \\ \mathbf{H}'\mathbf{Y} \end{pmatrix}, \tag{17}$$

so the solution to Eq. (17) can be computed just by inverting a matrix,

$$\begin{pmatrix} \boldsymbol{\beta}^{(1)} \\ \boldsymbol{\beta}^{(2)} \\ \vdots \\ \boldsymbol{\beta}^{(s)} \end{pmatrix} = \begin{pmatrix} \mathbf{H}'\mathbf{H}+C\mathbf{I} & \mathbf{H}'\mathbf{H} & \dots & \mathbf{H}'\mathbf{H} \\ \mathbf{H}'\mathbf{H} & \mathbf{H}'\mathbf{H}+C\mathbf{I} & \dots & \mathbf{H}'\mathbf{H} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{H}'\mathbf{H} & \mathbf{H}'\mathbf{H} & \dots & \mathbf{H}'\mathbf{H}+C\mathbf{I} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{H}'\mathbf{Y} \\ \mathbf{H}'\mathbf{Y} \\ \vdots \\ \mathbf{H}'\mathbf{Y} \end{pmatrix}. \tag{18}$$

This solution also works for simple BR with lineal regressors, replacing $\mathbf{H}'\mathbf{H}$ and $\mathbf{H}'\mathbf{Y}$ for $\mathbf{X}'\mathbf{X}$ and $\mathbf{X}'\mathbf{Y}$ respectively.

Generalised global boosting ridge ELM.

The generalisation is as simple as making $\mathbf{H}$ different for each s-th base learner,

$$\begin{pmatrix} \boldsymbol{\beta}^{(1)} \\ \boldsymbol{\beta}^{(2)} \\ \vdots \\ \boldsymbol{\beta}^{(s)} \end{pmatrix} = \begin{pmatrix} \mathbf{H}^{(1)'}\mathbf{H}^{(1)}+C\mathbf{I} & \mathbf{H}^{(1)'}\mathbf{H}^{(2)} & \dots & \mathbf{H}^{(1)'}\mathbf{H}^{(S)} \\ \mathbf{H}^{(2)'}\mathbf{H}^{(1)} & \mathbf{H}^{(2)'}\mathbf{H}^{(1)}+C\mathbf{I} & \dots & \mathbf{H}^{(2)'}\mathbf{H}^{(S)} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{H}^{(S)'}\mathbf{H}^{(1)} & \mathbf{H}^{(S)'}\mathbf{H}^{(2)} & \dots & \mathbf{H}^{(S)'}\mathbf{H}^{(S)}+C\mathbf{I} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{H}^{(1)'}\mathbf{Y} \\ \mathbf{H}^{(2)'}\mathbf{Y} \\ \vdots \\ \mathbf{H}^{(S)'}\mathbf{Y} \end{pmatrix}. \tag{19}$$

The different nonlinear feature mappings in $\mathbf{H}^{(s)}$ can be generated by any ELM method: randomisation[12], PCA with different subsets of the training dataset[44], elements in a pseudorandom sequence[45], $\cdots$ As mentioned before, with this generalisation, several random weights and biases have been selected for each mapping function $\mathbf{h}^{(s)}$ mappings, thus generating different mappings $\mathbf{H}^{(s)}$.

Methodology flowcharts.

Figure 1 includes a graphical and minimalistic comparison of the methodologies involved in this paper. Note that ELM (a) trains one model in a single step, BRELM and GBRELM (b) train several models sequentially, and the proposed GGBRELM (c) trains all models in a single step, since BRELM, GBRELM and GGBRELM are ensemble methodologies.

Analysis of the computational burden.

The ELM model's computational complexity is determined by the number of hidden nodes, denoted as D, the size of the training set, denoted as N, and the number of classes, J. To compute $\mathbf{H}'\mathbf{H}$, it is needed to multiply a matrix of $D \times N$ by a $N \times D$ resulting in a complexity of $O(D \cdot N^2)$. Then, ELM must perform matrix inversion on a $D \times D$ matrix whose complexity is $O(D^3)$ as shown in[46,47]. After that, a multiplication of the $\mathbf{H}'\mathbf{Y}$, that is, $D \times N$ by $N \times J$ with a cost of $O(D \cdot N \cdot J)$. Finally, the resulting matrices $D \times D$ and $D \times J$ are multiplied with a computational time of $O(D^2 \cdot J)$. Therefore, the total computational complexity is $O(\text{ELM}) = O(D \cdot N^2 + D^3 + D \cdot N \cdot J + D^2 \cdot J)$.

The computational cost for the BRELM and GBRELM methods also depends on the number of base learners S. Since these methodologies train S ELM models sequentially and each model is trained using the residual from the previous one as targets, the computational cost will be $O(S \cdot O(\text{ELM}) + (S-1)(N \cdot D \cdot J))$.

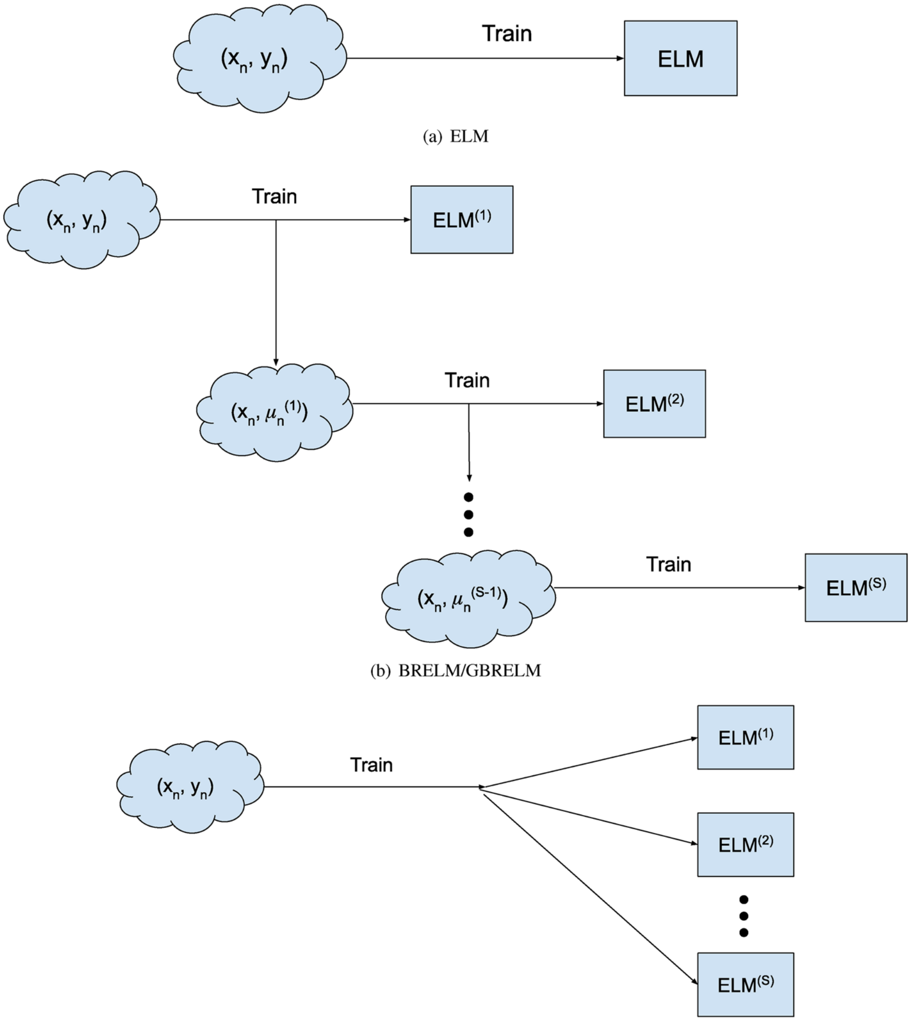
Finally, considering that GGBRELM performs optimisation in a single step, the method must calculate a matrix inversion of a $DS \times DS$ matrix and multiply the result with a $DS \times NJ$ matrix. Given that the $\mathbf{H}'\mathbf{H}$ matrix is symmetric, the computation of all the intermediate $\mathbf{H^{s'}}\mathbf{H^{t}}$ for $s = 1, \dots, S$, $t = s, \dots, S$, a total of $S(S-1)/2$ multiplications of matrices $D \times N$ by $N \times D$ need to be performed, resulting in a complexity of $O(S(S-1)/2 \cdot D \cdot N^2)$. For this reason, the computational cost of GGBRELM is $O(S(S-1)/2 \cdot D \cdot N^2 + (DS)^3 + (DS)^2 \cdot J + DS \cdot N \cdot J)$.

Experimental design

In order to evaluate the methodology presented in "Methodology of the proposal" Section, a comprehensive experimental environment has been implemented. In this sense, "Experiments" Section describes the experiments performed initially. "Datasets" Section includes a description of the datasets employed in the regression and classification problems. "Algorithms and parameters setting" Section contains a concise explanation of the algorithms selected for performing the comparative study and the set-up of their hyperparameters. Finally, the metrics implemented for the evaluation of the models are detailed in "Measures" Section, and the statistical tests carried out to validate the obtained results are defined in "Statistical tests" Section.

Experiments.

As stated before, the aim of this work is not only to improve the performance of the base learner (ELM) but also to overcome the disadvantages of the BRELM and, specifically, Generalised BRELM



(a) ELM

(b) BRELM/GBRELM

(c) GGBRELM

Figure 1. Minimalistic flowcharts of the different methodologies.

(GBRELM). Also, for comparison purposes, a recent kernel methodology is used (KBRELM, see Algorithms and parameters setting" Section). For this purpose, two experiments have been carried out:

- In the first experiment (E1), the number of neurons in the hidden layer was low. Thus, the smaller the number of hidden nodes, the worse ELM performs; on the other hand, GBRELM performs better.
- In the second experiment (E2), the number of nodes in the hidden layer is larger. Thereby, the performance capabilities of the ELM are high (strong learners), so this model achieves competitive results. At the same time, the GBRELM ensemble cannot take advantage of its ensemble architecture to improve its performance. As a classic ensemble, its performance increases when weak learners are used and decreases when complex learners are used.

In both experiments, the performance of the methodologies in the datasets will be analysed according to their size. Also, for the classification problems, the number of classes and the imbalance ratio, calculated as the ratio resulting from dividing the number of patterns of the majority class by the number of patterns of the minority class, will be examined.

The underlying idea is to demonstrate that GGBRELM outperforms ELM, GBRELM and KBRELM in both experimental scenarios by comparing them in regression and classification problems and performing an analysis according to different dataset properties.

Datasets. Experimental validation has been performed on 71 classification datasets and 52 regression datasets, respectively. This selection was carried out to include in the reference datasets various types of classification/regression problems in terms of their field of application, their size (product of the number of patterns times the number of attributes), their number of classes, and their imbalanced ratio. Tables 1 and 2 show a summary of the main characteristics of the selected datasets: identification number (*ID*), which has been assigned by

Classification datasets							
ID	**Dataset**	**#Inst.**	**#Attr.**	**Size**	**#Classes**	**Class distribution**	**IR**
Large datasets (Size > 100000)							
1	Weight-exercises	39242	53	2079826	5	(11159 7593 7214 6844 6432)	1.73
2	Cnae-9	1080	856	924480	9	(120 120 120 120 120 120 120 120 120)	1.00
3	Mushroom	8124	111	901764	2	(4208 3916)	1.07
4	Skin-segmentation	245057	3	735171	2	(194198 50859)	3.82
5	Statlog-shuttle	43500	9	391500	7	(34108 6748 2458 132 37 11 6)	5684.67
6	Letter-recognition	20000	16	320000	26	(813 805 803 796 792 789 787 786 783 783 775 773 768 766 764 761 758 755 753 752 748 747 739 736 734 734)	1.11
7	Spambase	4601	57	262257	2	(2788 1813)	1.54
8	Optical-recognition-handwritten-digits	3823	64	244672	10	(389 389 387 387 382 380 380 377 376 376)	1.03
9	Weight-lifting-exercises	4024	54	217296	5	(1370 1365 901 276 112)	12.23
10	Magic-gamma-telescope	19020	10	190200	2	(12332 6688)	1.84
11	Statlog-project-landsat-satellite	4435	36	159660	6	(1072 1038 961 479 470 415)	2.58
12	Ozone-level-detection-one	1848	72	133056	2	(1791 57)	31.42
13	Ozone-level-detection-eight	1847	72	132984	2	(1719 128)	13.43
14	Wall-following-robot-navigation-24	5456	24	130944	4	(2205 2097 826 328)	6.72
15	Chess-king-rook-vs-king-pawn	3196	38	121448	2	(1669 1527)	1.09
16	Electrical-grid	10000	12	120000	2	(6380 3620)	1.76
17	Pen-based-recognition-handwritten-digits	7494	16	119904	10	(780 780 780 779 778 720 720 719 719 719)	1.08
Medium datasets (10000 < Size < 100000)							
18	Thyroid-disease-ann-thyroid	3772	21	79212	3	(3488 191 93)	37.51
19	Thyroid-disease-allhyper	2800	27	75600	4	(2723 62 8 7)	389.00
20	Thyroid-disease-sick-euthyroid	3163	20	63260	2	(2870 293)	9.80
21	Hill-valley	606	100	60600	2	(305 301)	1.01
22	Hill-valley-noise	606	100	60600	2	(307 299)	1.03
23	Statlog-project-german-credit	1000	59	59000	2	(700 300)	2.33
24	Seismic-bumps	2584	22	56848	2	(2414 170)	14.20
25	Thyroid-disease-dis	2028	28	56784	2	(1989 39)	51.00
26	Thyroid-disease-sick	2028	28	56784	2	(1866 162)	11.52
27	Thyroid-disease-allbp	2028	23	46644	5	(936 716 265 82 29)	32.28
28	Qsar-biodegradation	1055	41	43255	2	(699 356)	1.96
29	Horse-colic	300	121	36300	2	(191 109)	1.75
30	Car-evaluation	1728	21	36288	4	(1210 384 69 65)	18.62
31	Libras-movement	360	90	32400	15	(24 24 24 24 24 24 24 24 24 24 24 24 24 24 24)	1.00
32	Credit-approval	666	46	30636	2	(367 299)	1.23
33	Tic-tac-toe-endgame	958	27	25866	2	(626 332)	1.89
34	Congressional-voting-records	435	48	20880	2	(267 168)	1.59
35	Breast-cancer-wisconsin-diagnostic	569	30	17070	2	(357 212)	1.68
36	Thoracic-surgery	470	27	12690	2	(400 70)	5.71
37	Connectionist-bench-sonar	208	60	12480	2	(111 97)	1.14
38	Dermatology	358	34	12172	6	(111 71 60 48 48 20)	5.55
39	Ionosphere	351	34	11934	2	(225 126)	1.79
40	Yeast	1484	8	11872	10	(463 429 244 163 51 44 35 30 20 5)	92.60
41	Breast-cancer	286	39	11154	2	(218 68)	3.21
42	Wall-following-robot-navigation-2	5456	2	10912	4	(2205 2097 826 328)	6.72
Small datasets (Size < 10000)							
43	Connectionist-bench	990	10	9900	11	(90 90 90 90 90 90 90 90 90 90 90)	1.00
44	Climate-model-simulation-crashes	540	18	9720	2	(494 46)	10.74
45	Teaching-assistant-evaluation	151	54	8154	3	(52 50 49)	1.06
46	Heart-disease-cleveland	299	23	6877	5	(161 54 36 35 13)	12.38
47	Breast-cancer-wisconsin-prognostic	194	32	6208	2	(148 46)	3.22
48	Breast-cancer-wisconsin	683	9	6147	2	(444 239)	1.86
Continued							

Classification datasets							
ID	Dataset	#Inst.	#Attr.	Size	#Classes	Class distribution	IR
49	Indian-liver-patient	579	10	5790	2	(414 165)	2.51
50	Heart-disease-hungarian	294	19	5586	2	(188 106)	1.77
51	Parkinsons	195	21	4095	2	(147 48)	3.06
52	Image-segmentation	210	19	3990	7	(30 30 30 30 30 30 30)	1.00
53	Spectf-heart	80	44	3520	2	(40 40)	1.00
54	Blood-transfusion-service-center	748	4	2992	2	(570 178)	3.20
55	Monks-problems-2	432	6	2592	2	(290 142)	2.04
56	Balance-scale	625	4	2500	3	(288 288 49)	5.88
57	Wine	178	13	2314	3	(71 59 48)	1.48
58	Planning-relax	182	12	2184	2	(130 52)	2.50
59	Soybean-small	47	45	2115	4	(17 10 10 10)	1.70
60	Glass-identification	214	9	1926	6	(76 70 29 17 13 9)	8.44
61	Hepatitis	80	19	1520	2	(67 13)	5.15
62	Seeds	210	7	1470	3	(70 70 70)	1.00
63	Thyroid-disease-new-thyroid	215	5	1075	3	(150 35 30)	5.00
64	Haberman-survival	306	3	918	2	(225 81)	2.78
65	Fertility	100	9	900	2	(88 12)	7.33
66	Monks-problems-1	124	6	744	2	(62 62)	1.00
67	Monks-problems-3	122	6	732	2	(62 60)	1.03
68	Balloons-a	20	4	80	2	(12 8)	1.50
69	Balloons-b	20	4	80	2	(12 8)	1.50
70	Balloons-c	20	4	80	2	(12 8)	1.50
71	Balloons-d	16	4	64	2	(9 7)	1.29

Table 1. Characteristics of the selected classification datasets, sorted by size.

ordering the datasets from the highest to the lowest size, name (*Dataset*), number of instances (*#Inst.*), attributes (*#Attr.*) and size (*Size*). According to their size, databases have been divided into large (size > 100000), medium (10000 < size < 100000) and small (size < 10000). The number of classes (*#Classes*), their distribution (*Class distribution*) and the imbalanced ratio (*IR*) have also been included in the characterisation of the classification problem datasets (Table 1). Imbalanced datasets (IR > 2) have also been underlined for further analysis. From here to the end, the datasets are annotated according to their ID. While classification datasets are extracted from UCI Machine Learning Repository[48], regression benchmark problems come from different machine learning repositories: UCI, Department of Statistics in the University of Florida[49] and LIACC[50].

Algorithms and parameters setting. The proposed method has been evaluated by comparing its results with respect to other recent state-of-the-art ELM proposals. The comparison methods are briefly described below:

- Extreme Learning Machine (ELM)[12] (described in "Extreme learning machine" Section). In the model implementation, the weights and bias in the hidden layer were randomly generated following a uniform distribution. In contrast, the output weights were optimised using the ELM minimisation problem with L_2 regularisation.
- Generalised BRELM (GBRELM) (a version combining the algorithm described in "Boosting ridge extreme learning machine" Section with the generalisation of mapping functions $\mathbf{h}^{(s)}$). This work compares the generalised version of Boosting Ridge for Extreme Learning Machine since it introduces variability into the model. Thus it would not make sense to compare with a simpler version where all ensemble elements have the same input layer.
- Generalised Global BRELM (GGBRELM) (described in Section "Methodology of the proposal"). The proposed methodology improves the sequential Generalised Boosting Ridge original architecture with a global approach.
- Kernel BRELM (KBRELM)[39]. In order to compare our proposal with a more recent methodology in the literature, we have also added a Boosting Ridge ensemble using as base learners Kernel Ridge Regression, as in[39]. This method works as the sequential Boosting Ridge for ELM presented in "Boosting ridge regression" Section but uses kernel trick instead of neural mapping. For it, Gaussian kernel was used, with hyperparameter γ,

$$\mathbf{k}(x_1, x_2) = \exp^{-\frac{\|x_1 - x_2\|^2}{\gamma}}.$$

Regression datasets					
ID	**Dataset**	**#Inst.**	**#Attr.**	**Size**	**Repository**
Large datasets (Size > 100000)					
1	News-popularity	39644	58	2299352	UCI ML repository
2	Casp	45730	9	411570	UCI ML repository
3	Friedman	40768	9	366912	LIACC (University of Porto)
4	Ailerons	7154	40	286160	LIACC (University of Porto)
5	Nwp-min	7590	21	159390	UCI ML repository
6	Elevators	8752	17	148784	LIACC regression repository
7	Electrical-grid	10000	12	120000	UCI ML repository
Medium datasets (10000 < Size < 100000)					
8	Parkinsons-motor	5875	16	94000	UCI ML repository
9	Parkinsons-total	5875	16	94000	UCI ML repository
10	Skillcraft	3338	18	60084	UCI ML repository
11	Winequality-white	4898	11	53878	UCI ML repository
12	Cpu-performance	209	245	51205	UCI ML repository
13	Abalone	4177	10	41770	UCI ML repository
14	Student-performance-por	649	43	27907	UCI ML repository
15	Parkinsons-speech	1040	26	27040	UCI ML repository
16	Usopen-men-2013a	126	168	21168	UCI ML repository
17	Usopen-men-2013b	126	168	21168	UCI ML repository
18	Frenchopen-men-2013a	123	170	20910	UCI ML repository
19	Frenchopen-men-2013b	123	170	20910	UCI ML repository
20	Wimbledon-women-2013a	118	170	20060	UCI ML repository
21	Wimbledon-women-2013b	118	170	20060	UCI ML repository
22	Wimbledon-men-2013a	113	163	18419	UCI ML repository
23	Wimbledon-men-2013b	113	163	18419	UCI ML repository
24	Winequality-red	1599	11	17589	UCI ML repository
25	Frenchopen-women-2013a	111	155	17205	UCI ML repository
26	Frenchopen-women-2013b	111	155	17205	UCI ML repository
27	Student-performance-mat	395	43	16985	UCI ML repository
28	Forestfires	517	28	14476	UCI ML repository
29	Ausopen-men-2013a	103	138	14214	UCI ML repository
30	Ausopen-men-2013b	103	138	14214	UCI ML repository
31	Ausopen-women-2013a	99	141	13959	UCI ML repository
32	Ausopen-women-2013b	99	141	13959	UCI ML repository
33	Triazines	186	60	11160	LIACC regression repository
Small datasets (Size < 10000)					
34	Automobile	160	62	9920	UCI ML repository
35	Usopen-women-2013a	74	106	7844	LIACC (University of Porto)
36	Airfoil-self-noise	1503	5	7515	LIACC (University of Porto)
37	Housing	506	13	6578	UCI ML repository
38	Auto-mpg	392	7	2744	UCI ML repository
39	Servo	167	12	2004	UCI ML repository
40	Pyrim	74	27	1998	UCI ML repository
41	Yatch	308	6	1848	UCI ML repository
42	Hybrid	153	11	1683	Departament of Statistics (University of Florida)
43	Lpga2009	146	11	1606	Departament of Statistics (University of Florida)
44	Brazilian-logistic	60	20	1200	UCI ML repository
45	Slump	103	7	721	UCI ML repository
46	Slump-flow	103	7	721	UCI ML repository
47	Slump-mpa	103	7	721	UCI ML repository
48	Japanemg	45	5	225	Departament of Statistics (University of Florida)
49	Beer	23	7	161	Departament of Statistics (University of Florida)
50	Const-maint	33	4	132	Departament of Statistics (University of Florida)
51	Diabetes	43	2	86	LIACC (University of Porto)
52	Texas-jan-temp	16	3	48	Departament of Statistics (University of Florida)

Table 2. Characteristics of the selected regression datasets, sorted by size.

The performance of the comparison methods depends critically on the setting of two hyperparameters: the regularisation parameter, C, and the number of hidden nodes, D. The hyperparameter C was determined by a grid search in a 5-fold nested cross-validation. The optimal value of the regularisation parameter for all comparison methods was determined with the following grid: $C \in \{10^{-2}, 10^{-1}, 1, 10, 10^{2}\}$. The number of hidden nodes, D, in all models was set to $D = 10$ for the first experiment and $D = 1000$ for the second one. In the case of the KBRELM method, the γ parameter needs to be crossvalidated, so it has been determined with the grid $\gamma \in \{10^{-2}, 10^{-1}, 1, 10, 10^{2}\}$. The ensemble size for all the ensemble methods was set to 10 base learners.

The experimental results were obtained using a 10-fold cross-validation procedure, with 3 repetitions per fold. Thus, 30 error measures were obtained for all methods compared, ensuring adequate statistical significance of the results. The partitions were the same for all models compared. Input values were standardised, regression labels were scaled to [0, 1] and class labels were binarised, following "1-to-*J*" encoding[51].

Measures. The metrics used for performance validation were all standard metrics in their environments, that is, well-known and standard metrics for classification and regression problems. In this regard, the simplicity and success of applying the accuracy rate (*Acc*) have allowed it to be widely used as a performance measure for classification problems. However, the *Acc* is unsuitable for imbalanced datasets, which is one of the big tradeoffs when using the accuracy metric. As seen in Table 1, there are a total of 35 datasets with an *IR* higher than 2, which is the threshold value considered in this work. Therefore, it is more appropriate to use balanced accuracy (*Balanced Accuracy*), which is equal to the accuracy in balanced datasets and considers the imbalance of classes when it exists. In addition, two other classification metrics, Precision (*Precision*) and F-measure (*F*1), have also been used because they are useful in balanced and imbalanced scenarios.

Given a binary classification problem (positives and negatives patterns), it is considered:

- True positives (*TP*): positive patterns predicted as positive.
- False negatives (*FN*): positive patterns predicted as negative.
- False positives (*FP*): false patterns predicted as positive.
- True negative (*TN*): false patterns predicted as negative.

Then, these classification performance metrics are mathematically defined as follows:

- *Balanced Accuracy* is the mean of Sensitivity and Specificity. Imbalanced datasets can be addressed by using the average of Sensitivity and Specificity. If a model only predicts accurately for the majority class in the dataset, it will receive a worse *Balanced Accuracy* score:

$$Balanced\ Accuracy = \frac{Sensivity + Specificity}{2} = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right). \quad (20)$$

- *Precision* is the percentage of positive patterns predicted as positive with respect to the total of positive predicted patterns:

$$Precision = \frac{TP}{TP+FP}. \quad (21)$$

- *F*1 is the harmonic mean of the Precision and Recall:

$$F1 = 2\frac{Precision \cdot Recall}{Precision + Recall} = \frac{2TP}{2TP+FP+FN}. \quad (22)$$

For multi-class problems, the metrics are calculated by comparing one class against all the others. The chosen class is considered positive, while the others are negative. This approach allows for obtaining a metric value for each of the classes. Then, the mean value is obtained.

The root mean square error (*RMSE*) and the determination coefficient (R^2) are the principal measures in the validation of an algorithm for regression problems:

- *RMSE* is the standard deviation of the differences between predicted and target values, and it is defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(\hat{\mathbf{y}}(\mathbf{x}_n) - y_n\right)^2}, \quad (23)$$

  where $\hat{\mathbf{y}}(\mathbf{x}_n)$ is the predicted value for pattern $\mathbf{x}_n$, and y_n, the real one.
- R^2 is the determination coefficient representing the proportion of the variation in the dependent variable that is predictable from the independent variables.

$$R^2 = \frac{\sigma^2_{\hat{\mathbf{y}},\mathbf{y}}}{\sigma^2_{\hat{\mathbf{y}}}\sigma^2_{\mathbf{y}}}, \quad (24)$$

  where $\mathbf{y}$ and $\hat{\mathbf{y}}$, are the real and predicted values, respectively.

Statistical tests. In order to demonstrate that the GGBRELM model is a promising method in its field, it is crucial to validate its performance with respect to that of the comparison methods with statistical tests. For both experiments and for each metric, a pre-hoc test was applied with the evaluations of the methods on the different datasets to assess the statistical significance of the rank differences. For evaluations where the test detected statistical differences in method rankings, a post-hoc test was conducted to determine which models are distinctive among the multiple comparisons performed using the best performing method as the control method. For this purpose, nonparametric tests were applied. First, nonparametric Friedman's tests[52], with *Balanced Accuracy*, *Precision* and *F*1 (classification), and *RMSE* and R^2 (regression) ranking of the models as test variables, were carried out for $\alpha = 0.05$. Then, nonparametric Holm's post-hoc test[53] was implemented to determine whether the control method, the GGBRELM, statistically outperforms the comparison methods considering $\alpha = 0.05$ and taking into account each metric.

Discussion of the results

This section includes the analysis of the experimental results obtained on the selected datasets. This part of the paper has been divided into two sections according to classification and regression datasets. For the sake of conciseness, it has been opted to provide only the relevant graphs and a summary of the statistical results.

Classification datasets. The generalisation performances of the considered methods for E1 ($D = 10$) and E2 ($D = 1000$) in classification datasets are shown in Figs. 2 and 3, respectively ((a) *Balanced Accuracy*, (b) *Precision*, (c) *F*1). In those figures, the Y-axis represents the value of the reported metric, while the X-axis contains the IDs of the datasets sorted by size. If GGBRELM is the best for one dataset, its ID appears in bold, and if it is the second best, it appears in italics. Finally, imbalanced datasets are marked with an underline. For the case of the all classification metrics, the higher the point is located on the graph, the better performance of that method since the objective is to maximise these metrics.

As a general rule, it can be observed that the GGBRELM methodology outperforms the other approaches in *Balanced Accuracy*, *Precision* and *F*1 in both experiments. Significantly, the difference is greater in those datasets where all the methodologies do not achieve good performances.

In particular, in E1, when comparing *Balanced Accuracy*, GGBRELM performs better in 31 datasets, and it is the second best in 36, representing almost the total number of databases. For precision, it is the best in 36 datasets and the second one in 30. Moreover, for the F1, GGBRELM is also the best in 36 datasets and the second in 27. GBRELM and KBRELM have similar performance regarding the number of databases in which they are the best or second. ELM performance is lower than the ensemble approaches, according to the literature.

Furthermore, in experiment E2, where the classifiers are configured with a high number of neurons in the hidden layer, the ELM becomes more specialised. Hence its performance improves, and it should outperform the ensemble methods due to its disadvantages when using strong base learners, such as saturation or overfitting. Nevertheless, while it is true that GBRELM and KBRELM obtain worse results than ELM, GGBRELM overcomes this disadvantage of ensemble nature methods by getting more accurate results. Thus, GBBRELM achieves the best result in 27, 30 and 28 datasets in terms of *Balanced Accuracy*, *Precision* and *F*1, respectively, and the second best in 31, 30 and 30 datasets. Thus, the proposed methodology is also better than the three compared methods, as shown in Fig. 3.

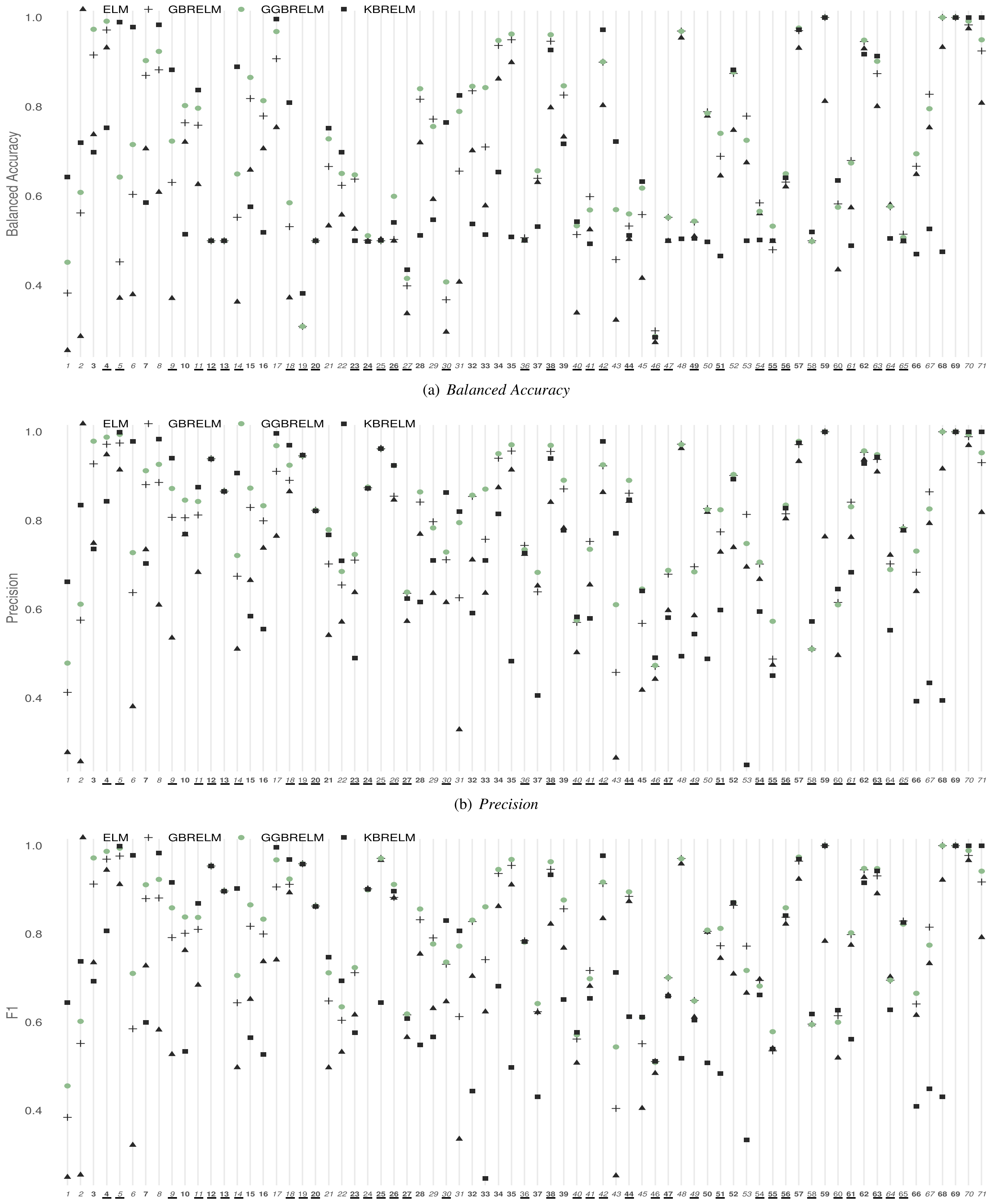
As mentioned above, a set of statistical tests have been carried out to analyse the results from statistical hypothesis contrasts, summarising the results in Table 3. For the Friedman's tests and a level of significance $\alpha = 5\%$, the confidence interval is $C_0 = (0, F_{0.05} = 2.65)$, and the F-distribution statistical value considering *Balanced Accuracy* rankings is $F^* = 27.80$, considering *Precision* rankings is $F^* = 31.69$ and taking into account *F*1 is $F^* = 22.73$ in the experiment E1 (D = 10), while in the E2 experiment (D = 1000), $F^* = 15$, $F^* = 10.76$ and $F^* = 9.89$, respectively. Consequently, in both experiments, the test rejects the null-hypothesis stating that all algorithms perform equally in mean ranking of *Balanced Accuracy*, *Precision* and *F*1. That is, the algorithm effect is statistically significant. For this reason, it is considered the best performing method as a control method for a post-hoc test, comparing this algorithm with the rest of the methods. In this way, Table 3 also shows the results of Holm's test. When using GGBRELM as the control algorithm (CA), Holm's test shows that $p_i < \alpha_i^*$ in all cases, for $\alpha = 0.05$, confirming that there are statistically significant differences favouring GGBRELM in both experiments and for each metric.

Discussion considering dataset size. As aforementioned, the datasets have been sorted in decreasing order of size and have been divided into three categories according to it, as shown in Table 1: 17 large datasets (*IDs* 1-17), 25 medium (*IDs* 18-42) and 29 small ones (*IDs* 43-71).

Looking at E1, for large datasets, GGBRELM is the best in 8 datasets and the second in the remaining ones for all metrics. It is the best in 12, 13 and 13 medium datasets and the second in 11, 10 and 9 according to *Balanced Accuracy*, *Precision* and *F*1, respectively. For small datasets, the best results are achieved on 11, 15 and 15, and the second best on 16, 11 and 9 datasets, depending on the metric analysed.

For the case of E2, for large datasets, GGBRELM is the best in 11, 10 and 9 and the second best in 4, 6 and 7. For medium datasets, the best are obtained in 6, 10 and 9, while the second best results are achieved in 14, 11 and 10. Finally, the best results are obtained in 10 and the second best in 13 small datasets in all metrics.

As can be seen, regardless of size, the GGBRELM method performs quite well. However, for both E1 and E2, the best results are concentrated in the large datasets being the best or second best method in almost all metrics in both experiments. In the smallest datasets, the improvement of the proposal is not as noticeable as in the remaining ones. It makes sense since they are databases without difficulty and are easier to solve by any method.

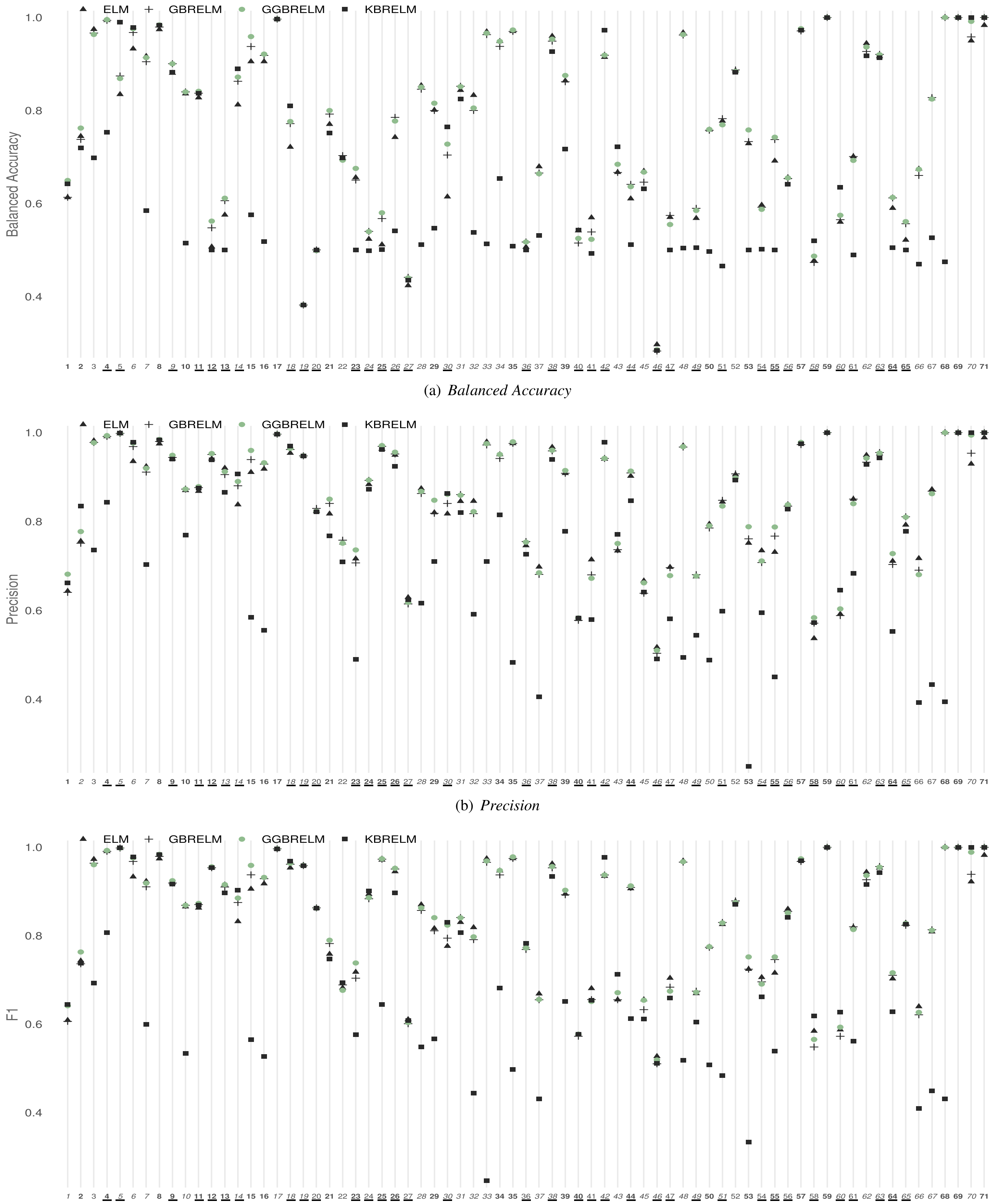


(a) *Balanced Accuracy*

(b) *Precision*

(c) *F*1

Figure 2. Performance plot on metrics for classification datasets using D = 10. The Y-axis represents the value of the metric, while the X-axis contains the IDs of the datasets sorted by size. If GGBRELM is the best for that dataset, its ID appears in bold, and if it is the second best, it appears in italics. Finally, imbalanced datasets are marked with an underline.



(a) *Balanced Accuracy*

(b) *Precision*

(c) *F*1

Figure 3. Performance plot on metrics for classification datasets using D = 1000. The Y-axis represents the value of the metric, while the X-axis contains the IDs of the datasets sorted by size. If GGBRELM is the best for that dataset, its ID appears in bold, and if it is the second best, it appears in italics. Finally, imbalanced datasets are marked with an underline.

		Experiment 1		Experiment 2	
Balanced sccuracy	**Friedman**	$C_0 = (0, F_{0.05} = 2.65)$	$F^* = 27.80(\ast)$	$C_0 = (0, F_{0.05} = 2.65)$	$F^* = 15(\ast)$
	Holm	**CA:GGBRELM**		**CA:GGBRELM**	
i	$\alpha^*_{0.05}$	**Algorithm**	p_i	**Algorithm**	p_i
1	0.017	ELM	0.0000 (*)	KBRELM	0.0000 (*)
2	0.025	KBRELM	0.0002 (*)	GBRELM	0.0063 (*)
3	0.050	GBRELM	0.0148 (*)	ELM	0.0375 (*)
Precision	**Friedman**	$C_0 = (0, F_{0.05} = 2.65)$	$F^* = 31.69(\ast)$	$C_0 = (0, F_{0.05} = 2.65)$	$F^* = 10.76(\ast)$
	Holm	**CA:GGBRELM**		**CA:GGBRELM**	
i	$\alpha^*_{0.05}$	**Algorithm**	p_i	**Algorithm**	p_i
1	0.017	ELM	0.0000 (*)	KBRELM	0.0000 (*)
2	0.025	KBRELM	0.0015 (*)	GBRELM	0.0015 (*)
3	0.050	GBRELM	0.0085 (*)	ELM	0.0052 (*)
F1	**Friedman**	$C_0 = (0, F_{0.05} = 2.65)$	$F^* = 22.73(\ast)$	$C_0 = (0, F_{0.05} = 2.65)$	$F^* = 9.89(\ast)$
	Holm	**CA:GGBRELM**		**CA:GGBRELM**	
i	$\alpha^*_{0.05}$	**Algorithm**	p_i	**Algorithm**	p_i
1	0.017	ELM	0.0000 (*)	KBRELM	0.0000 (*)
2	0.025	KBRELM	0.0000 (*)	GBRELM	0.0005 (*)
3	0.050	GBRELM	0.0177 (*)	ELM	0.0027 (*)

Table 3. Results of the Friedman's and Holm's tests using GGBRELM as control algorithm (CA) when comparing its average *Balanced Accuracy*, *Precision* and *F*1 to those of ELM, GBRELM and KBRELM: corrected α values, compared methods and p values, all of them ordered by the number of comparison (i). CA results statistically better than the compared algorithm are marked with (*).

Discussion considering imbalanced datasets. In the experimental validation, there are a total of 35 imbalanced datasets. As stated, for each classification database, the *IR* has been calculated as the ratio of the number of patterns in the majority class to the number of patterns in the minority class. The *IR* has been reported in Table 1, underlining those datasets with an $IR > 2$. In addition, in Figs. 2 and 3, the *IDs* of these imbalanced datasets have also been underlined, making it easier to discuss the results by taking them into account.

Considering the first experiment with *D* set to 10, GGBRELM achieves the best result on 13 datasets and the second best on 18, resulting in almost the total number of databases, considering the *Balanced Accuracy* metric. Similar is what happens with the other two metrics, being the best in 15 and second best in 15 for *Precision* and obtaining the best results in 16 and second best in 11 with *F*1. In this case, it is worth noting that the second method would be GBRELM on average for the three metrics. Although KBRELM obtains the best result in many databases, this showed an unstable behaviour since it is either very good or the worst, depending on the dataset.

As for E2, the same happens for GGBRELM, being the best method for the three metrics in 9, 13 and 12 datasets, respectively, and the second best method in 18, 16 and 13. It is important to note that for imbalanced datasets, the GBRELM method has approximately the same average performance in all metrics with respect to ELM, but ELM is still slightly better than GBRELM.

From this analysis, it can be concluded that the proposed GGBRELM method not only performs well on all metrics for all databases but is also the most appropriate for imbalanced datasets.

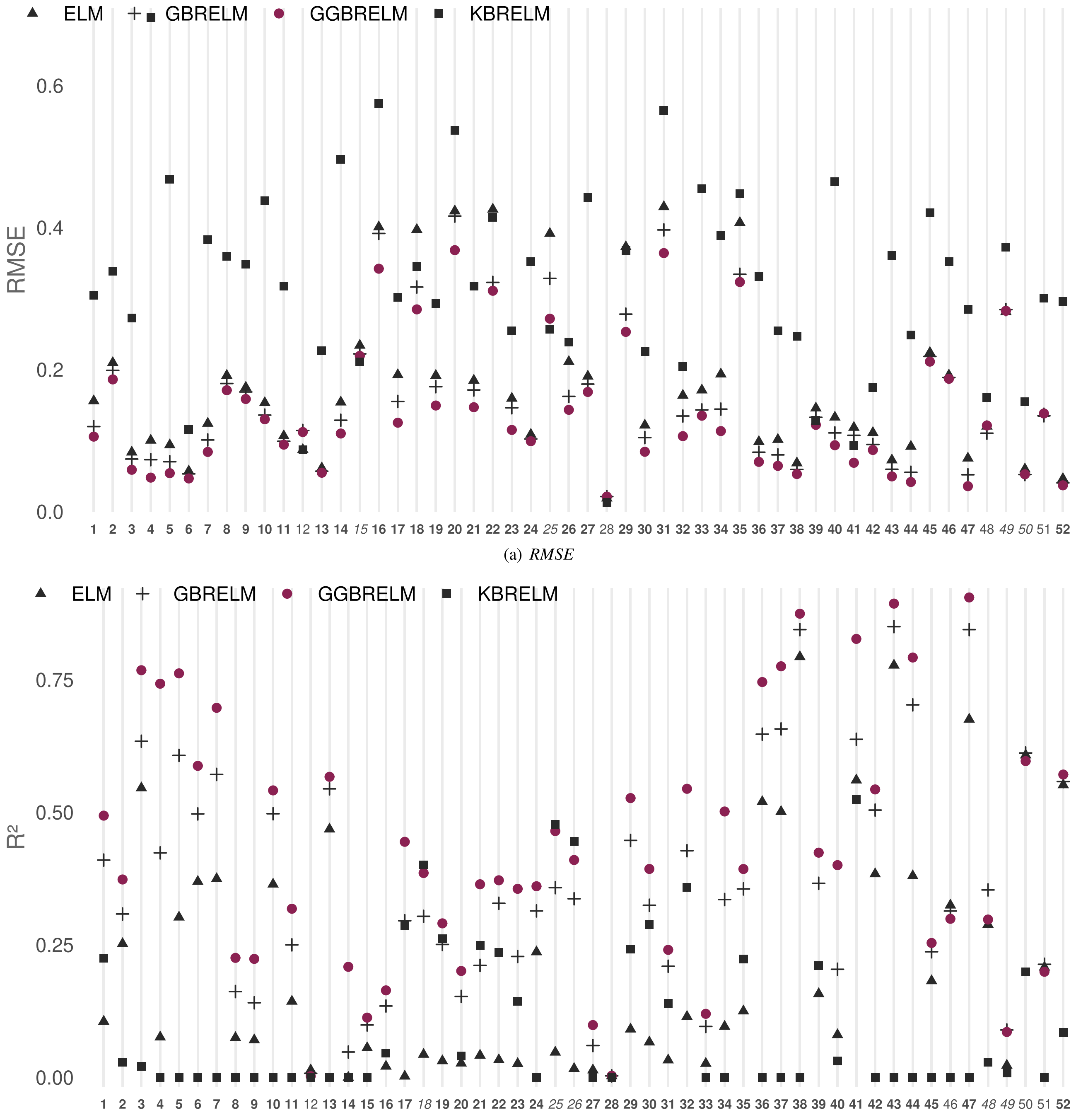
Discussion considering the number of classes. From column *#Classes* in Table 1 and Figs. 2 and 3, the influence of the number of classes on the results obtained can be analysed.

Thus, for E1 and the 44 binary problems, GGBRELM is the best algorithm on average since it is the best on 26, 27 and 28 databases depending on the analysed metric (*Balanced Accuracy*, *Precision* and *F*1). In addition, it is the second best on 16, 14 and 11, respectively. In the case of multiclass problems, and specifically as the number of classes increases, KBRELM performs similarly to GGBRELM in this experiment. This can be explained by the fact that the higher the number of classes, the more complex the problem becomes, and the algorithms with a higher number of connections benefit, as is the case of kernels.

However, for the case of E2, i.e., when GGBRELM is provided with more neurons in its base classifiers, the results indicate that it performs better on average than the rest of the algorithms in binary and multiclass problems in all metrics. Thus, in binary problems, GGBRELM is the best in 20, 22 and 21, and the second in 14, 13 and 13, respectively. For the case of problems with a more significant number of classes, it is the best in 7, 8 and 7 and the second best in practically the remaining ones, making it the best algorithm on average.

Regression datasets.

The performances of the considered methods for E1 ($D = 10$) and E2 ($D = 1000$) in regression datasets are shown in Figs. 4 and 5, respectively ((a) *RMSE*, (b) R^2). As in classification datasets, the Y-axis represents the value of the reported metric, while the X-axis contains the IDs of the datasets sorted by size. If GGBRELM is the best for one dataset, its ID appears in bold, and if it is the second best, it appears in italics. For the case of the *RMSE* metric, the lower the point is located on the graph, the better performance of



(a) *RMSE*

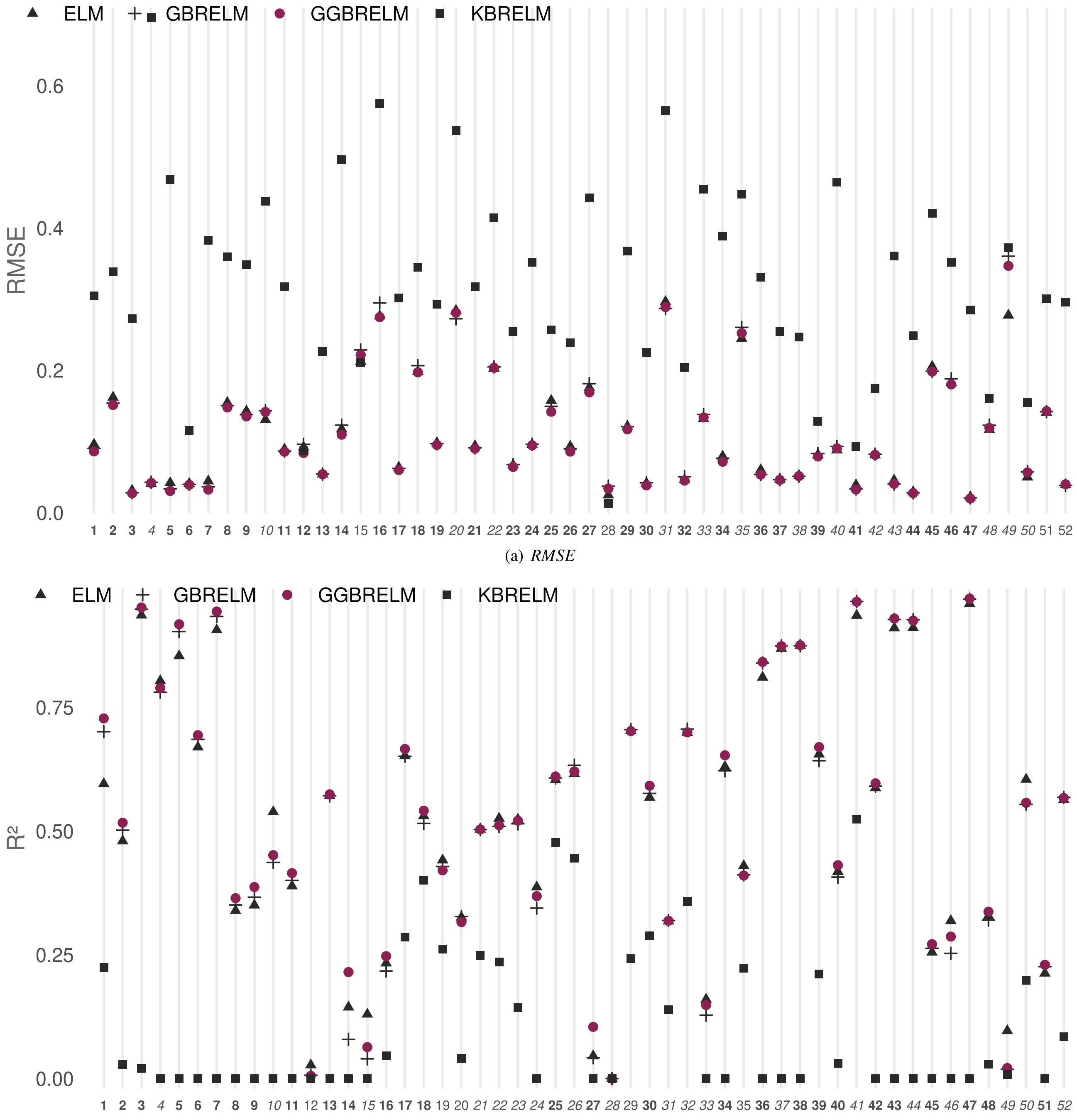
(b) R^2

Figure 4. Performance plot on metrics for regression datasets using D = 10. The Y-axis represents the value of the metric, while the X-axis contains the IDs of the datasets sorted by size. If GGBRELM is the best for that dataset, its ID appears in bold, and if it is the second best, it appears in italics.

that method since the objective is to minimise this metric. The opposite occurs in the R^2 metric because it must be maximised.

The findings unambiguously demonstrate that the GGBRELM methodology outperforms the alternative approaches in both experiments and across both metrics. This distinction is especially evident in datasets where the other methodologies exhibit suboptimal performance.

Thus, in the case of E1, GGBRELM is the best method in 44 datasets and the second best in 4 datasets in terms of *RMSE*. In addition, it is the best method in 43 datasets and the second best in 5 datasets when comparing R^2. With a low number of neurons, GBRELM also outperforms ELM since it is a weak learner. However, KBRELM does not seem to perform well in problems of this nature, being the worst regressor of the four methods.



(a) *RMSE*

(b) R^2

Figure 5. Performance plot on metrics for regression datasets using D = 1000. The Y-axis represents the value of the metric, while the X-axis contains the IDs of the datasets sorted by size. If GGBRELM is the best for that dataset, its ID appears in bold, and if it is the second best, it appears in italics.

Furthermore, in experiment E2, GGBRELM overcomes the disadvantage of ensemble nature methods by getting more accurate results regarding *RMSE* and R^2. Hence, GGBRELM achieves the better *RMSE* performance in 34 datasets and the second best in 14. Similarly, it gets the best R^2 in 28 datasets and the second best in 19.

In the same way, as in classification datasets, four Friedman's tests have been run showing the rejection of the null-hypothesis given that, for $\alpha = 5\%$, the confidence interval is $C_0 = (0, F_{0.05} = 2.66)$, and the statistical values for *RMSE* and R^2 are $F^* = 102.63$ and $F^* = 101.97$ in E1, and $F^* = 77.21$ and $F^* = 91.05$ in E2 (Table 4). This Table also shows the results of Holm's test comparing *RMSE* and R^2. Again, when using GGBRELM as the control algorithm (CA), Holm's test shows that $p_i < \alpha_i^*$ in all cases, for $\alpha = 0.05$, confirming that there are statistically significant differences favouring GGBRELM in both experiments and metrics.

		Experiment 1		Experiment 2	
RMSE	Friedman	$C_0 = (0, F_{0.05} = 2.66)$	$F^* = 102.63(*)$	$C_0 = (0, F_{0.05} = 2.66)$	$F^* = 77.21(*)$
	Holm	CA:GGBRELM		CA:GGBRELM	
i	$\alpha^*_{0.05}$	Algorithm	p_i	Algorithm	p_i
1	0.017	KBRELM	0.0000 (*)	KBRELM	0.0000 (*)
2	0.025	ELM	0.0000 (*)	GBRELM	0.0001 (*)
3	0.050	GBRELM	0.0005 (*)	ELM	0.0004 (*)
R^2	Friedman	$C_0 = (0, F_{0.05} = 2.66)$	$F^* = 101.97(*)$	$C_0 = (0, F_{0.05} = 2.66)$	$F^* = 91.05(*)$
	Holm	CA:GGBRELM		CA:GGBRELM	
i	$\alpha^*_{0.05}$	Algorithm	p_i	Algorithm	p_i
1	0.017	KBRELM	0.0000 (*)	KBRELM	0.0000 (*)
2	0.025	ELM	0.0000 (*)	GBRELM	0.0063 (*)
3	0.050	GBRELM	0.0024 (*)	ELM	0.0185 (*)

Table 4. Results of the Friedman's and Holm's tests using GGBRELM as control algorithm (CA) when comparing its average *RMSE* and R^2 to those of ELM, GBRELM and KBRELM: corrected α values, compared methods and p-values, all of them ordered by the number of comparison (i). CA results statistically better than the compared algorithm are marked with (*).

Discussion considering dataset size. The regression datasets have been ordered from the highest to the smallest size and have also been divided into three categories as shown in Table 2: 7 large datasets (*IDs* 1-7), 26 medium (*ID* 8-33) and 29 small (*ID* 34-52).

Considering E1, for large datasets, GGBRELM is the best in all datasets for all metrics. For medium size, it is the best in 22 in both metrics and the second in 2 and 3, respectively. For small datasets, the best results are achieved on 15 and 14, and the second best on 2 datasets in both metrics.

For the case of E2, for large datasets, GGBRELM is the best in 6 datasets and the second in 1 for both metrics. For medium datasets, the best are obtained in 19 and 11, while the second best results are obtained in 5 and 11. Finally, for small datasets, the best are obtained in 9 and 11, and the second best in 8 and 7 datasets.

In both experiments, the dataset size does not influence since, in all cases, the GGBRELM algorithm is much better than the others. However, it can be observed how in the five smallest databases, the performance difference of GGBRELM with respect to the other methods decreases since they lack complexity and are susceptible to being solved with any method.

Conclusions

This paper presents a new ensemble methodology that tackles the problem of base learners saturation and a drop in performance when strong base learners are used in the ensemble method, avoiding increase iteratively the size of the ensemble. To solve this, this method performs a global optimisation in the Boosting Ridge methodology, using Extreme Learning Machine models as base learners. The proposed ensemble method, Generalised Global Boosting Ridge for Extreme Learning Machine, generates a set of initial input layer mappings with different parameters for their hidden layers. The output layer weights are optimised in one step, reducing the generalisation error of the ensemble.

A complete experimentation has been carried out, taking into account 71 classification datasets, analysing their size, the number of classes and the imbalance ratio, and 52 regression datasets considering their size, all from different application domains. The experiments show that i) the proposed Generalised Global ensemble method for ELM outperforms Generalised Boosting Ridge in different contexts, that is, low number and high number of neurons, and ii) Generalised Global methodology improves the results of ELM when it is specialised with a high number of neurons, overcoming the disadvantage of ensemble methods in these scenarios. Instead of relying on generating diversity through weak learners (low number of neurons), our method depends on its optimisation in the final prediction of the ensemble as a whole, thus not relying on the implicit diversity of the hidden neurons mapping.

In future work, it planned to adapt the ensemble learning framework to other base learners and other machine learning paradigms, such as ordinal regression or semisupervised learning. And finally, the application of the methodology to real-world problems could be proposed.

Data availability

The databases used together with the code necessary for their extraction are available at https://github.com/cperales/uci-download-process. The code generated in the experimental design, including the proposed methodology is available at https://github.com/cperales/pyridge. The whole table results obtained during the current study are available from the corresponding author upon reasonable request.

Received: 21 March 2023; Accepted: 18 July 2023
Published online: 21 July 2023

References

1. Huang, G.-B., Zhu, Q.-Y. & Siew, C.-K. Extreme learning machine: Theory and applications. *Neurocomputing* **70**, 489–501 (2006).
2. Zhang, Y. *et al.* Multi-kernel extreme learning machine for EEG classification in brain-computer interfaces. *Expert Syst. Appl.* **96**, 302–310 (2018).
3. Pan, Z., Meng, Z., Chen, Z., Gao, W. & Shi, Y. A two-stage method based on extreme learning machine for predicting the remaining useful life of rolling-element bearings. *Mech. Syst. Signal Process.* **144**, 106899 (2020).
4. Zuo, E. *et al.* Rapidly detecting fennel origin of the near-infrared spectroscopy based on extreme learning machine. *Sci. Rep.* **12**, 13593 (2022).
5. Khan, M. A. *et al.* Prediction of covid-19-pneumonia based on selected deep features and one class kernel extreme learning machine. *Comput. Electr. Eng.* **90**, 106960 (2021).
6. She, Q., Zou, J., Meng, M., Fan, Y. & Luo, Z. Balanced graph-based regularized semi-supervised extreme learning machine for EEG classification. *Int. J. Mach. Learn. Cybern.* **12**, 903–916 (2021).
7. Sattar, A. M., Ertuğrul, Ö. F., Gharabaghi, B., McBean, E. A. & Cao, J. Extreme learning machine model for water network management. *Neural Comput. Appl.* **31**, 157–169 (2019).
8. Ali, M. *et al.* Coupled online sequential extreme learning machine model with ant colony optimization algorithm for wheat yield prediction. *Sci. Rep.* **12**, 5488 (2022).
9. Huang, G.-B., Zhou, H., Ding, X. & Zhang, R. Extreme learning machine for regression and multiclass classification. *IEEE Trans. Syst. Man. Cybern. Part B (Cybernetics)* **42**, 513–529 (2011).
10. Hecht-Nielsen, R. Theory of the backpropagation neural network. In *Neural Networks for Perception* 65–93 (Elsevier, USA, 1992).
11. De Chazal, P., Tapson, J. & Van Schaik, A. A comparison of extreme learning machines and back-propagation trained feed-forward networks processing the mnist database. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 2165–2168. (IEEE, 2015).
12. Huang, G.-B., Zhou, H., Ding, X. & Zhang, R. Extreme learning machine for regression and multiclass classification. *IEEE Trans. Syst. Man Cybern. B Cybern.* **42**, 513–29 (2012).
13. Miche, Y. *et al.* Op-elm: Optimally pruned extreme learning machine. *IEEE Trans. Neural Netw.* **21**, 158–162 (2009).
14. Yang, Y. & Duan, Z. An effective co-evolutionary algorithm based on artificial bee colony and differential evolution for time series predicting optimization. *Complex Intell. Syst.* **6**, 299–308 (2020).
15. Li, L., Qi, S., Zhou, H. & Wang, L. Prediction of line heating deformation on sheet metal based on an ISSA-ELM model. *Sci. Rep.* **13**, 1252 (2023).
16. Khellal, A., Ma, H. & Fei, Q. Ensemble of extreme learning machines for regression. In *2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)* 1052–1057. (IEEE, 2018).
17. Zhai, J., Zhang, S., Zhang, M. & Liu, X. Fuzzy integral-based elm ensemble for imbalanced big data classification. *Soft. Comput.* **22**, 3519–3531 (2018).
18. Song, G. & Dai, Q. A novel double deep elms ensemble system for time series forecasting. *Knowl. Based Syst.* **134**, 31–49 (2017).
19. Zou, W., Yao, F., Zhang, B. & Guan, Z. Improved meta-elm with error feedback incremental elm as hidden nodes. *Neural Comput. Appl.* **30**, 3363–3370 (2018).
20. Raghuwanshi, B. S. & Shukla, S. Classifying imbalanced data using ensemble of reduced kernelized weighted extreme learning machine. *Int. J. Mach. Learn. Cybern.* **10**, 3071–3097 (2019).
21. Kumar, N. K., Savitha, R. & Al Mamun, A. Ocean wave height prediction using ensemble of extreme learning machine. *Neurocomputing* **277**, 12–20 (2018).
22. Chen, Z., Jiang, C. & Xie, L. A novel ensemble elm for human activity recognition using smartphone sensors. *IEEE Trans. Ind. Inf.* **15**, 2691–2699 (2018).
23. Chen, H., Tan, C. & Lin, Z. Ensemble of extreme learning machines for multivariate calibration of near-infrared spectroscopy. *Spectrochim. Acta Part A Mol. Biomol. Spectrosc.* **229**, 117982 (2020).
24. Xie, S. *et al.* Birdsongs recognition based on ensemble elm with multi-strategy differential evolution. *Sci. Rep.* **12**, 9739 (2022).
25. Krogh, A. *et al.* Neural network ensembles, cross validation, and active learning. *Adv. Neural. Inf. Process. Syst.* **7**, 231–238 (1995).
26. Breiman, L. Bagging predictors. *Mach. Learn.* **24**, 123–140 (1996).
27. Freund, Y. & Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **55**, 119–139 (1997).
28. Schaal, S. & Atkeson, C. G. From isolation to cooperation: An alternative view of a system of experts. In *Advances in Neural Information Processing Systems* 605–611. (NIPS, 1996).
29. Bühlmann, P. & Yu, B. Boosting with the l2 loss: Regression and classification. *J. Am. Stat. Assoc.* **98**, 324–339 (2003).
30. Singhal, Y., Jain, A., Batra, S., Varshney, Y. & Rathi, M. Review of bagging and boosting classification performance on unbalanced binary classification. In *2018 IEEE 8th International Advance Computing Conference (IACC)* 338–343. (IEEE, 2018).
31. Ko, A. H., Sabourin, R., De Oliveira, L. E. & De Souza Britto, A. The implication of data diversity for a classifier-free ensemble selection in random subspaces. In *19th International Conference on Pattern Recognition* 2251–2255. (ICPR, 2008).
32. Tutz, G. & Binder, H. Boosting ridge regression. *Comput. Stat. Data Anal.* **51**, 6044–6059 (2007).
33. Kodahl, A. R. *et al.* Novel circulating microRNA signature as a potential non-invasive multi-marker test in ER-positive early-stage breast cancer: a case control study. *Mol. Oncol.* **8**, 874–883 (2014).
34. Binder, H. & Schumacher, M. Incorporating pathway information into boosting estimation of high-dimensional risk prediction models. *BMC Bioinform.* **10**, 1–11 (2009).
35. Tollenaar, N. & van der Heijden, P. G. M. Optimizing predictive performance of criminal recidivism models using registration data with binary and survival outcomes. *PLoS ONE* **14**, 1–37 (2019).
36. Brown, G., Wyatt, J. L. & Tiňo, P. Managing diversity in regression ensembles. *J. Mach. Learn. Res.* **6**, 1621–1650 (2005).
37. Cai, Y., Liu, X., Zhang, Y. & Cai, Z. Hierarchical ensemble of extreme learning machine. *Pattern Recogn. Lett.* **116**, 101–106 (2018).
38. Xue, X., Yao, M., Wu, Z. & Yang, J. Genetic ensemble of extreme learning machine. *Neurocomputing* **129**, 175–184. https://doi.org/10.1016/j.neucom.2013.09.042 (2014).
39. Lin, S.-B., Lei, Y. & Zhou, D.-X. Boosted kernel ridge regression: Optimal learning rates and early stopping. *J. Mach. Learn. Res.* **20**, 1738–1773 (2019).
40. Sun, T. & Zhou, Z.-H. Structural diversity for decision tree ensemble learning. *Front. Comput. Sci.* **12**, 560–570 (2018).
41. Dietterich, T. G. Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems* 1–15 (Springer, 2000).
42. Ran, Y. *et al.* Boosting ridge extreme learning machine. In *Proceedings—2012 IEEE Symposium on Robotics and Applications, ISRA 2012* 881–884 (2012).
43. Deng, W., Zheng, Q. & Chen, L. Regularized extreme learning machine. In *2009 IEEE Symposium on Computational Intelligence and Data Mining* 389–395. (IEEE, 2009).
44. Castaño, A., Fernández-Navarro, F. & Hervás-Martínez, C. PCA-ELM: A robust and pruned extreme learning machine approach based on principal component analysis. *Neural Process. Lett.* **37**, 377–392 (2013).
45. Cervellera, C. & Macciò, D. Low-discrepancy points for deterministic assignment of hidden weights in extreme learning machines. *IEEE Trans. Neural Netw. Learn. Syst.* **27**, 891–896 (2015).

46. Cook, S. A. An overview of computational complexity. *Commun. ACM* **26**, 400–408 (1983).
47. Durán-Rosal, A. M., Durán-Fernández, A., Fernández-Navarro, F. & Carbonero-Ruz, M. A multi-class classification model with parametrized target outputs for randomized-based feedforward neural networks. *Appl. Soft Comput.* **133**, 109914 (2023).
48. Dua, D. & Graff, C. UCI machine learning repository (2017).
49. Winner, L. Miscellaneous datasets (2020).
50. Torgo, L. Regression datasets (2020).
51. Harris, D. *Digital Design and Computer Architecture* (Elsevier/Morgan Kaufmann, Amsterdam, 2012).
52. Friedman, M. A comparison of alternative tests of significance for the problem of m rankings. *Ann. Math. Stat.* **11**, 86–92 (1940).
53. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **7**, 1–30 (2006).

Acknowledgements

This work was fully supported by the research funds from "Universidad Loyola Andalucía".

Author contributions

C.P.G. and A.M.D.R. processed the experimental data; C.P.G. performed the analysis and the implementation; J.P.R. designed the figures; A.M.D.R. and J.P.R. were involved in planning and supervising the work; all authors wrote and reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to J.P.-R.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.