Data Mining and Knowledge Discovery (2024) 38:461–500
https://doi.org/10.1007/s10618-023-00957-1

Optimal selection of benchmarking datasets for unbiased machine learning algorithm evaluation

João Luiz Junho Pereira[1] · **Kate Smith-Miles**[2] · **Mario Andrés Muñoz**[3] · **Ana Carolina Lorena**[1]

Received: 9 February 2023 / Accepted: 6 July 2023 / Published online: 20 October 2023

Abstract

Whenever a new supervised machine learning (ML) algorithm or solution is developed, it is imperative to evaluate the predictive performance it attains for diverse datasets. This is done in order to stress test the strengths and weaknesses of the novel algorithms and provide evidence for situations in which they are most useful. A common practice is to gather some datasets from public benchmark repositories for such an evaluation. But little or no specific criteria are used in the selection of these datasets, which is often ad-hoc. In this paper, the importance of gathering a diverse benchmark of datasets in order to properly evaluate ML models and really understand their capabilities is investigated. Leveraging from meta-learning studies evaluating the diversity of public repositories of datasets, this paper introduces an optimization method to choose varied classification and regression datasets from a pool of candidate datasets. The method is based on maximum coverage, circular packing, and the meta-heuristic Lichtenberg Algorithm for ensuring that diverse datasets able to challenge the ML algorithms more broadly are chosen. The selections were compared experimentally with a random selection of datasets and with clustering by *k*-medoids and proved to be more effective regarding the diversity of the chosen benchmarks and the ability to challenge the ML algorithms at different levels.

Keywords Benchmark datasets' suites · Instance space analysis · Classification algorithms · Regression algorithms · Meta-learning · Optimization

Responsible editor: Charalampos Tsourakakis.

Extended author information available on the last page of the article

1 Introduction

A common practice in Machine Learning (ML) studies is to evaluate one or more algorithms on various datasets. However, the no-free lunch theorem (Wolpert 2002) states that all ML algorithms are likely to perform equally well on average if we consider all classes of problems they can be applied to. So applying a set of benchmarks necessitates an understanding of whether they are indeed representing all classes of problems, or are biased in some way that makes clear which algorithms are better suited to those biased classes. The characterization of the conditions under which any algorithm is expected to outperform others is an important task for algorithm selection (Luengo and Herrera 2015), but relies on being able to control the diversity of selected benchmarks.

Despite the presence of a large number of datasets in public repositories such as UCI (Dua and Graff 2017), OpenML (Vanschoren et al. 2014), Keel (Alcalá-Fdez et al. 2011), and competitions, the selection of the datasets comprising a benchmark set for evaluating new ML solutions is usually random, or based on simple criteria such as limiting the number of observations or input features considered. This raises doubts as to whether the strengths and weaknesses of the compared ML algorithms are likely to be revealed, or if all algorithms will appear similar on average due to the lack of criteria to ensure their diversity and explore a full range of biases.

The few studies found proposing benchmark datasets are more concerned with their application field than their general aspects or properties: such as in materials (Clement et al. 2020), semantic web (Ristoski et al. 2016), website phishing detection (Hannousse and Yahiouche 2021), physical systems (Takamoto et al. 2022), graphs (Hu et al. 2020), and atmospheric sciences (Dueben et al. 2022). Olson et al. (2017) and Bischl et al. (2017) are among the only ones to consider more general aspects by selecting a subset of 165 and 100 classification datasets, respectively. The first work manually curated datasets from public repositories and analysed their properties based on a set of meta-features values and the algorithmic performance of some ML classifiers. And Bischl et al. (2017) selected their datasets by applying 13 rules on thousands of datasets from the OpenML repository. These rules target datasets with some desired properties (such as number of observations, features, imbalance ratio, among others) and exclude datasets that are ill-conditioned or that are too easy (defined as having maximum 10-fold cross-validation accuracy when solved by a decision tree). A study proposing an optimization method to select a benchmark of diverse datasets able to challenge the ML algorithms at different levels from a predefined pool of candidate datasets has never been done before. Regarding regression problems, the literature is even more limited (Muñoz et al. 2021).

The Meta-learning (MtL) (Vanschoren 2019) area provides multiple ways to characterize a dataset, which can be explored in order to assess inner properties from the data. Indeed, an active field in MtL is the proposal of meta-features for extracting different properties from the datasets. Among them, one can cite statistical, information-theoretic, landmarking, model-based, and complexity measures (Lorena et al. 2018, 2019; Rivolli et al. 2022). Therefore, much more information can be used to support the choice of diverse datasets for evaluating new

ML solutions, an area which remains under explored. This is one of the main problems to be addressed in this paper: how to assemble an optimized benchmark of labeled datasets for evaluating new ML models and solutions based on meta-knowledge about their properties?

This paper proposes a methodology to select a subset of diverse and challenging benchmark datasets which can stress-test the domains of competence of supervised ML algorithms in an unbiased manner. Both classification and regression problems are considered. For such, the meta-knowledge extracted in previous work of the literature (Munoz et al. 2018; Muñoz et al. 2021) and made publicly available at an online platform named Melbourne Algorithm Test Instance Library with Data Analytics (MATILDA) https://matilda.unimelb.edu.au/matilda/) is used.

We build this work from the framework of Instance Space Analysis (ISA), developed by Smith-Miles and co-authors over many years, and summarized as a methodology in Smith-Miles and Muñoz (2023). In recent years, ISA has been applied to understanding strengths and weaknesses of algorithms in many fields, and to evaluate the diversity of existing benchmark suites. Munoz et al. (2018) and Muñoz et al. (2021) present an ISA of classification and regression problems in ML where datasets are placed in a 2-dimensional instance space showing linear trends regarding their difficulty level according to different criteria. A set of meta-features is used to assess their properties and to obtain the projections. This paper proposes an optimization method to solve the benchmark selection problem based on the maximum coverage of this latent space, so that diversity according to different meta-features values can be taken into account directly in this choice. Recently, Alipour et al. (2023) proposed an strategy for selecting subsets of instances that retain diversity across the instance space by ensuring uniform density and discarding redundant instances, which was applied to the analysis of maximum flow problems. Our work differs from the previous proposal by formulating a geometric maximum coverage optimization strategy for covering the instance space, while also controlling the number of datasets to be selected as desired. The following criteria are addressed in our proposed approach:

- **(C1) Stress-testing the choice of diverse benchmarks for unbiased evaluation of algorithms**: whenever an ad-hoc selection takes place, one may bias the results obtained. Ad-hoc selection can sometimes appear to have facilitated cherry picking of datasets for which better algorithm results are obtained, while bad results are simply omitted. But we argue one needs to know and understand situations where the new algorithm succeeds and fails as well. This practice is needed to improve the ML field as a whole and to make the evaluation of new solutions less biased, more comprehensive and trustworthy.
- **(C2) Taking advantage of meta-knowledge on public repositories**: there is plenty of previous work extracting meta-knowledge from public repositories employed in ML studies (Soares 2009; Macià and Bernadó-Mansilla 2014; Bischl et al. 2017; Munoz et al. 2018; Muñoz et al. 2021). There is need to leverage such knowledge to support Data Scientists in different ways. Here the objective is to support practitioners in choosing their test cases for better coverage of different situations that may challenge their algorithms;

- **(C3) Formulating the choice of diverse benchmarks as an optimization problem**: given a set of N datasets, the goal is to choose subsets $M < N$ with definite number (the operator chooses how many datasets) which is diverse according to different characteristics and that can challenge ML algorithms at different levels. This set should contain both easy, medium, and hard to predict datasets, with different characteristics and structural properties. But other subsets might be preferred, such as only a hard set of datasets. The method is general and can easily accommodate such preferences. This is an NP-hard maximum coverage combinatorial optimization problem combined with circle packing and can be solved using meta-heuristics.

The main **contributions** of this work are: (1) although the importance of a broad evaluation of new algorithms is well known in general Computer Science (Hooker 1995), there has been less focus in the ML field regarding this issue. This paper reinforces this concern when evaluating new ML solutions, not only in situations where an algorithm outperforms the others, but also to understand and characterize situations for which the algorithm does not perform so well; (2) acknowledging the existing literature on evaluating the diversity of public repositories of datasets in ML (Munoz et al. 2018; Macià and Bernadó-Mansilla 2014; Muñoz and Smith-Miles 2020; Muñoz et al. 2021), it is important to learn lessons and leverage new knowledge from this rich information. This paper makes use of some recent and state-of-the-art meta-knowledge reported in the literature in a new setting designed to support ML practitioners in evaluating their algorithmic solutions more properly; (3) given a search space represented as a latent bi-dimensional representation space where datasets are linearly distributed according to different meta-characteristics and the algorithmic performance of a pool of popular ML solutions, an optimization method based on maximum coverage and circle packing problems in this space is proposed. A recent and efficient meta-heuristic joining concurrently population and trajectory based search strategies inspired by lightning storms named Lichtenberg algorithm (LA) (Pereira et al. 2021b) is used to solve the problem.

The results obtained in experiments for both classification and regression problems demonstrates the importance of properly selecting diverse benchmarks of datasets for evaluating ML models. Our automatic method to recommend datasets with diverse properties aids such methodological design choice efficiently. Compared to a random selection of datasets and to a k-medoids selection (with $k = M$) applied in the 2D space, the diversity of datasets is ensured more properly with our method.

The remainder of this paper is organized as follows. Section 2 presents the main concepts involved in this study. The complete developed optimization methodology is presented in Sect. 3. The set of datasets selected in the experiments are analyzed in Sects. 4 and 5 draws conclusions.

2 Background

2.1 Meta-learning

The ML area has evolved and grown very quickly in the last decades, with the proposal of many new techniques and the expansion of the areas of application covered

(Davenport and Ronanki 2018). Controlled experimental design methodologies and standard benchmarks for testing such techniques have also been devised, increasing the reliability in their evaluation (Thiyagalingam et al. 2022). But it is still common to face challenges when deploying ML-based systems in practice (Paleyes et al. 2022), making it not uncommon that results become disappointing. This raises the question on whether current evaluation repositories and benchmarks are really representative so as to challenge the ML techniques at different levels, stressing their main capabilities but also limitations (Macià and Bernadó-Mansilla 2014; Munoz et al. 2018).

Meanwhile, the Metalearning (MtL) community has been studying how to leverage higher-level knowledge on previous problems already solved by ML techniques (Vanschoren 2019). This meta-knowledge, which can be extracted independently of the target domain addressed, has been used to support algorithmic selection for new problems with similar characteristics and properties (Smith-Miles 2009). Here we advocate such meta-knowledge can be useful beyond algorithmic selection and may also support a better evaluation of the strengths and weaknesses of ML techniques.

For such, this work takes advantage of the meta-knowledge gathered in some recent pieces of work in the MtL area. Munoz et al. (2018) and Muñoz et al. (2021) present an Instance Space Analysis (ISA) of classification and regression problems in ML, where datasets from popular repositories are analyzed regarding their capabilities to challenge different ML algorithms. A 2-dimensional space named Instance Space (IS) is built so that linear trends of the difficulty level of the datasets according to different perspectives are preserved.

Given a set $\mathcal{I}$ of N datasets, which can be gathered from public repositories commonly adopted by the ML community or from other sources, the ISA framework involves six main steps (Smith-Miles and Muñoz 2023): assembling a meta-dataset; constructing an instance space; generating ML predictions for automated algorithm selection; generating algorithm footprints, defined as areas of the IS where the algorithm is expected to perform well; analyzing the IS; and generating additional metadata if required. Next we present the first two steps, which allow to explain how the IS projections are obtained.

2.1.1 Assembling the meta-dataset

The meta-dataset must join two components: (i) properties of the base datasets that evidence their characteristics and difficulty levels; and (ii) the predictive performance of a pool of ML algorithms when applied to such datasets. Let $\mathcal{F}$ denote the set of meta-features and $\mathcal{A}$ denote the pool of algorithms considered, whose predictive performances on the datasets $\mathcal{I}$ are recorded in a set $\mathcal{Y}$.

Regarding the first component, there are many meta-features for characterizing classification datasets in the literature (Rivolli et al. 2022). For regression datasets, the literature is less abundant (Aguiar et al. 2022). Generally, they can be divided into the following categories:

- *Simple measures*: basic characteristics, such as the datasets size, their input features number, among others;

- *Statistical measures*: measures of localization, dispersion, distribution, and correlation of variables;
- *Information theoretic measures*: measures of the information content of the variables, as entropy and mutual information;
- *Model-based features*: take the characteristics of models built using the dataset, e.g. the size of decision trees;
- *Landmarking features*: consider the predictive performance of simple baseline models on the datasets, as the error rate of decision stumps;
- *Complexity/hardness measures*: capture the intrinsic difficulty in solving the problem by structural and geometric descriptors extracted from data.

Each dataset in $\mathcal{I}$ should be described by a set of metafeatures $\mathcal{F}$, which will characterize them. As a result, the meta-dataset will have a tabular format, with N rows (each dataset) and d columns (each meta-feature value for the given dataset).

In turn, the pool of algorithms $\mathcal{A}$ should include representatives with different biases. As each technique can extract distinct representations from data, they might suit better the structures of certain datasets and not others. The ultimate objective is to understand for which data conformations each technique can be recommended or not. The algorithms must receive the datasets in $\mathcal{I}$ as input and, by a cross-validation procedure, estimate the predictive performance attained when solving the underlying classification/regression problem using such data.

For classification problems, common performance metrics that can be used are Accuracy, AUC (Area Under the ROC Curve), Precision, Recall, F-measure, among others (Ferri et al. 2009). For regression problems, where the data labels are continuous, performance metrics taking into account the distance between the predicted values and the true labels are preferred, as the Mean Squared Error (MSE), Mean Absolute Error (MAE), etc. (Botchkarev 2018). By choosing one metric, the set $\mathcal{Y}$ containing the evaluation of the algorithms $\mathcal{A}$ in the datasets $\mathcal{I}$ can be assembled.

In ISA analysis, the set $\mathcal{Y}$ is concatenated to the set $\mathcal{F}$ to form the final meta-dataset $\mathcal{M}$ to be analysed, as illustrated in Fig. 1. $\mathcal{M}$ will have N rows and $d + a$ columns, where a is the number of algorithms included in the pool $\mathcal{A}$. In contrast, in standard MtL studies, each row of the meta-dataset is labeled according to either the best performing algorithm, a ranking of algorithms or the performance of a chosen algorithm (Garcia et al. 2018).

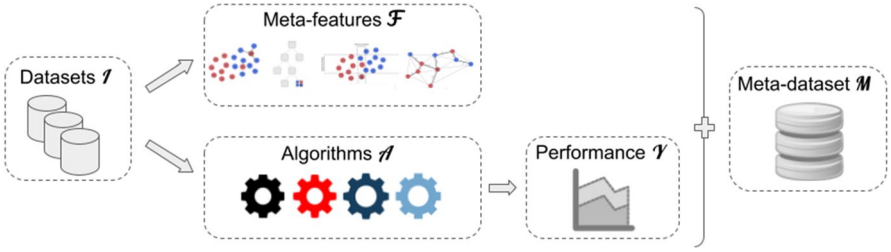


Fig. 1 Assembling the meta-dataset for ISA analysis

2.1.2 Obtaining the IS

Smith-Miles and Muñoz (2023) frame the IS obtainment as an optimization problem named *Projecting Instances with Linearly Observable Trends* (PILOT). The objective is to find an optimum mapping from the meta-dataset $\mathcal{M}$ to a 2-D instance space where linear trends of the meta-features values and of the performance metrics are observed. Before, $\mathcal{M}$ is subject to a feature selection process where only those meta-features which are more predictive of the algorithms' performance are kept.

Let $\mathbf{F} \in \mathbb{R}^{d \times N}$ be a matrix containing the meta-features values after feature selection for all datasets. Similarly, let $\mathbf{Y} \in \mathbf{R}^{N \times a}$ be a matrix containing the performances of the a algorithms on the same N datasets. The 2-D projection of the instances for this group of meta-features and algorithms is achieved by finding the matrices $\mathbf{A}_r \in \mathbb{R}^{2 \times d}$, $\mathbf{B}_r \in \mathbb{R}^{d \times 2}$, and $\mathbf{C}_r \in \mathbb{R}^{a \times 2}$ which minimize the following approximation error:

$$\|\mathbf{F} - \widehat{\mathbf{F}}\|_F^2 + \|\mathbf{Y} - \widehat{\mathbf{Y}}\|_F^2 \tag{1}$$

with:

$$\mathbf{Z} = \mathbf{A}_r \mathbf{F} \tag{2}$$

$$\widehat{\mathbf{F}} = \mathbf{B}_r \mathbf{Z} \tag{3}$$

$$\widehat{\mathbf{Y}}^{\top} = \mathbf{C}_r \mathbf{Z} \tag{4}$$

where $\mathbf{Z} \in \mathbb{R}^{2 \times N}$ is the matrix which gives the coordinates of the datasets (instances) in the 2-D space and $\mathbf{A}_r$ is the projection matrix mapping the meta-features values to the new space. Summarizing, the objective is to find the optimal linear transformation matrix $\mathbf{A}_r$, such that the mapping of all instances from $\mathbb{R}^d$ to $\mathbb{R}^2$ results in the strongest possible linear trends across the IS when inspecting the distribution of each algorithm's performance metric and of each of the meta-feature's values.

This optimization problem is rewritten as an alternative optimization problem assuming that $d < N$ and that $\mathbf{F}$ is full row rank and solved numerically using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm (Broyden 1970; Muñoz et al. 2021). From multiple runs of the algorithm, typically 30, the solution that achieves a maximum topological preservation is chosen (Yarrow et al. 2014). That is the solution showing maximum Pearson Correlation between the distances in the original meta-feature space and the distances in the IS.

The IS of classification datasets and of regression datasets from Munoz et al. (2018) and Muñoz et al. (2021) are presented in Fig. 2a, b, respectively. They are colored according to different meta-features' values: fraction of borderline points in the case of classification problems and number of observations in the case of regression problems. The higher the fraction of borderline points a training dataset has, the more complex the underlying classification problem tends to be, possibly requiring more complex decision boundaries to separate the classes. According to

(a) Classification (b) Regression

Fig. 2 IS of classification and regression datasets from Munoz et al. (2018) and Muñoz et al. (2021)

this observation, more complex datasets are placed towards the upper-right corner of the IS in Fig. 2a. In the case of the regression IS, datasets with a larger number of observations are placed in the upper-right corner of the IS. Taking the premise that more training data should lead to better predictive performance, in Fig. 2b the more complex datasets are in the down left region of the IS. More details on these IS are presented in Sect. 3. Although more dimensions could be considered when building the IS, the usage of two dimensions has a visual appeal, which also applies to our method. One can easily select regions of the IS to focus the search procedure, as done in our experiments where specific quadrants of the ISs are subject to the benchmark selection process.

Since the IS is a two-dimensional square geometric space, strategies can be adopted to select instances in order to build a benchmark of datasets that is as diverse as possible, with a definite number M inferior to the whole set, that is, $M < N$. It is important to emphasize to the reader that the IS used in this work is constantly being improved, either by adding new datasets (real or synthetic), meta-features, ML algorithms, ways of processing data, methodologies, among others (Smith-Miles and Muñoz 2023). Therefore, the results of this work are faithful to the current IS for classification and regression problems.

For the optimization process, an objective function is built based on two classic and complex NP-hard problems found in the literature: (i) the maximum coverage and (ii) the circular packing problems (Hochbaum 1996).

2.2 Maximum coverage problem

The *Maximum Coverage* (MC) is a combinatorial optimization problem that consists of finding a collection of sets $S = \{S_1, \ldots, S_M\}$ in a domain of elements $E = \{e_1, \ldots, e_N\}$ so that the collection of elements in $S_i \subseteq S$ is maximized, that is,

a maximum coverage is attained. From this original version, there are three other versions: (i) the weighted version adds weights $\{w_i\}_{i=1}^{N}$ to the elements and these weights must also be maximized, (ii) the budgeted version, in which each collection of sets S can be associated with $\{c_j\}_{i=1}^{M}$ costs and its sum must be less than a specified budget B, and (iii) generalized maximum coverage, which is a compostion of the previous versions (Khuller et al. 1999). The choice of one of these versions is determined by the problems' constraints and interests. The complete version is described in Eq. 5 (Cohen and Katzir 2008):

$$\max \sum_{e \in E_i, S_i} w_i(e_j) y_{ij} \tag{5}$$

Subject to:

$$\sum c_i(e_j) y_{ij} + \sum c(S_i) x_i \leqslant B_i \tag{6}$$

$$\sum_i y_{ij} \leqslant 1 \tag{7}$$

$$y_{ij} \in \{0, 1\} \tag{8}$$

$$x_i \in \{0, 1\} \tag{9}$$

where $y_{ij} = 1$ if e_j is covered by a set S_i and if $x_i = 1$, S_i is activated for the cover.

Even before the 2000s, this formulation has been applied for optimization in several areas, such as facility location, job scheduling, and circuit layout (Khuller et al. 1999). Until then, because it is a combinatorial optimization problem, the most used iterative and heuristic algorithm was the greedy algorithm (Zhang et al. 2000), that remains used nowadays. However, many limitations of this type of solution have been evidenced (Bang-Jensen et al. 2004) and new and more powerful algorithms have emerged to solve applications using the MC problem's formulation.

Several meta-heuristics have been successfully employed to solve the MC problem. Nascimento and Bastos-Filho (2010) applied the Particle Swarm Optimization (PSO) to solve the cellular base stations positioning using MC as objective function. Rahmani et al. (2018) applied a genetic algorithm (GA) and simulated annealing (SA) in the supply chain network design optimization, concluding that the first algorithm was more accurate. Another important area is the design of wireless sensor networks, in which Tossa et al. (2022) also used GA. For this same application, Taşdemir et al. (2022) compared the Immune Plasma Algorithm (IPA) and the Artificial Bee Colony (ABC) and concluded that the first had better results, but both showed significant improvements in relation to classical algorithms. Also recently, Matt et al. (2022) applied MC for the first time in literature in a ML problem. Using a nested GA, the authors proposed a method to extract a set of decision rules that best explains a classification data set.

2.3 Circular packing

Circular Packing (CP) is a problem related to MC, with some fundamental differences. This is also a complex optimization problem, consisting in finding how many equal circles can fit inside another geometric figure. Or in other words, considering a set C of M circles with location and size (x_i, y_i, r_i), what are the locations of these M circles that have the MC of the geometric figure considered? Note that there can be two approaches to find the C_M locations, considering for example that they are inscribed in a square: (i) Fix the radius circle and find M or (ii) Fix M and find the radius, which can be the same or different for each circle. In both cases the MC is achieved the closer the circles are and constraining that they do not overlap (Castillo et al. 2008).

Being also a combinatorial and NP-hard problem, several algorithms have been proposed to solve it. Again, the best solutions were obtained using meta-heuristics. Flores et al. (2016) used the GA, Diferential Evolution (DE), PSO, and the Evolution Strategy (ES) and concluded that the first two algorithms came closer of the exact solutions to CP into squares and other circles. A similar result was obtained by Yuan et al. (2022) using a hybrid GA-greedy algorithm.

Section 3 explains how MC and CP are composed here to guide a meta-heuristic to segregate the IS in a defined number of sets M, from which representative prototypes are chosen. But there is general a trend to apply new and improved meta-heuristics to get better results in solving these complex optimization problems. This paper uses a novel meta-heuristic inspired by lightning storms and Lichtenberg Figures, described next. Section A.3 of the Appendix presents and applies other three popular meta-heuristics from the literature to one instance of our problem.

2.4 Lichtenberg algorithm

The *Lichtenberg Algorithm* (LA) is a physics based meta-heuristic with mono and multi-objective versions (Pereira et al. 2021b, 2022). Both were successfully tested against traditional and recent meta-heuristics using complex groups of test functions. LA proved to be a promising algorithm overcoming traditional and recent optimization algorithms such as GA, PSO, ACO, NSGA-II, MOPSO, MOEA/D, MOGOA, and MOGWO. Since then, it has been efficiently applied in complex optimization problems as wind speed prediction optimizing Artificial Neural Networks models (Tian and Wang 2022), damage identification being treated as a minimization inverse method (Pereira et al. 2021a), subsurface imaging antenna capacitance (Janairo et al. 2022), structural design (Pereira et al. 2022b), image segmentation (Xiao and Cheng 2022; Ma et al. 2023), and sensor placement in helicopter main rotor blade (Pereira et al. 2022a). In the latter case the LA algorithm found optimal positions for a reduced number of sensors within more than 10,000 candidate nodes. This is a problem similar to the one addressed in the present work.

LA uses figures similar to lightning, with tortuous aspects similar to fractals, whose propagation has chaotic dynamics and are called Lichtenberg Figures (LF).

LA creates them using a stochastic cluster growth theory named Diffusion Limited Aggregation (Witten and Sander 1981) and shoots figures in the search space with random sizes and rotations at each iteration, centered on the best solution of the previous step. The LF is created as a cluster of particles (or points) and just a limited *Pop* number of them are used as population for evaluation in the objective function. The LF creation is fully numerical: a binary and squared matrix with size twice R_c (creation radius) is built like a map and in the center, a particle represented by the number one is fixed. The cluster is built by unitary values and the empty spaces worth zero.

Particles are randomly released across the matrix and if they reach the cluster, they have an *S* probability of fixing (stickiness coefficient), which controls the cluster's (or LF's) density. The particle can be added only if there is another particle next to it confirmed by a lateral check. If it reaches a radius slightly larger than the R_c, it is exterminated and another one starts the random walk again. This happens until all the particles (N_p) determined are contained in the cluster or until it reaches its construction limit. These three LA's parameters are about LF construction. Then, each particle can be transformed into locations on a Cartesian plane and the LF can be plotted at any size, slope or starting point. This process allows flat (2D) and spatial (3D) LF, which means that they could be used for problems of two and three decision variables. When dealing with more variables, a projection (or mirroring) of these figures is made.

Another optimizer's parameter is the refinement (*ref*), that can range from 0 to 1 and is a creator of a second LF (red) every iteration from zero to one hundred the size of the main LF (blue), see Fig. 3. This smaller scale figure improves exploitation by always having half of *pop*, when it exists (*ref* > 0)—see Fig. 3. *Pop* are black dots, usually set 10 times the number of design variables (*D*). They are randomly chosen throughout the LF structure (which is modified at each iteration). This unique hybrid routine not found in any meta-heuristics brought to the algorithm a great capacity for both exploitation and exploration.

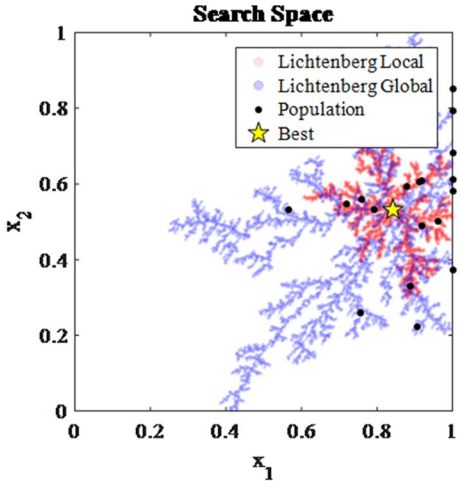


Fig. 3 LA in bidimensional search space (Color figure online)

The sixth parameter of the algorithm is the LF switching factor (M). It can be worth zero, one or two. If one, a LF is generated when starting the program and used in all iterations. If it is two, a new figure is generated and used at each iteration (implying in a huge computational cost). The fastest way is using $M = 0$, where a previously optimized LF is used in the optimizer: no figure is generated. Finally, the number of iterations (N_{iter}) is the algorithm's stopping criterion. Figure 14 summarizes the algorithm.

3 Methodology

The main objective of this work is to assemble a benchmark of M datasets that challenges ML algorithms in different ways, where M is set by the user. The IS presented in Fig. 2a has 235 classification datasets and the IS in Fig. 2b has 4885 regression datasets, both represented in a bidimensional square space with Cartesian coordinates. The closer one dataset is to another in this space, the closer their similarities in terms of difficulties as measured in the embedded space. First, the two previous ISs are described in more details, followed by the proposed optimization problem formulation.

3.1 IS of classification problems

The IS of classification datasets was built in Munoz et al. (2018) using 235 datasets (composing the set $\mathcal{I}$), where 210 are UCI instances (Dua and Graff 2017), 19 are Keel instances (Alcalá-Fdez et al. 2011), and 6 are DCol instances (Orriols-Puig et al. 2010). This collection of datasets has up to 11,055 observations and up to 1558 input features.

The pool of algorithms $\mathcal{A}$ has as representatives: Naive Bayes (NB), Linear Discriminant (LDA), Quadratic Discriminant (QDA), Classification and Regression Trees (CART), J48 decision tree (J48), k-Nearest Neighbor (KNN), Support Vector Machines with linear, polynomial and radial basis kernels (L-SVM, p-SVM, and RB-SVM, respectively), and Random Forests (RF). They are popular algorithms in ML and represent a wide range of biases. Their predictive performance registered in $\mathcal{Y}$ is measured by an error rate after running a ten-fold cross-validation procedure, as presented in Eq. 10 (Munoz et al. 2018):

$$ER = \frac{FN + FP}{n} \tag{10}$$

where FN is the number of false negatives and FP is the number of false positives and n is the size of the dataset.

Finally, the set of meta-features $\mathcal{F}$ is initially composed by 509 candidate meta-features, which are reduced to 10 after a meta-feature selection step. The meta-dataset $\mathcal{M}$ is then composed by 235 rows, described by 10 meta-features and with the error rate performance measures of 10 classifiers. After submitting this meta-dataset

to the PILOT, the resulting coordinates of the datasets are given as (Munoz et al. 2018):

$$\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} = \begin{bmatrix} 0.07 & 0.18 \\ 0.094 & 0.618 \\ -0.277 & -0.052 \\ 0.114 & 0.192 \\ 0.045 & -0.1 \\ -0.128 & 0.151 \\ -0.045 & 0.077 \\ 0.184 & 0.017 \\ 0.449 & 0.223 \\ 0.132 & -0.112 \end{bmatrix}^T \begin{bmatrix} H(X)_{max} \\ H(c) \\ \bar{M}_{CX} \\ DN_{ER} \\ SD(v) \\ F3 \\ F4 \\ L2 \\ N1 \\ N4 \end{bmatrix} \tag{11}$$

where:

- $H(X)_{max}$ is the maximum normalized entropy of the input features, quantifying the highest amount of information contained in the input features, assuming they are independent from each other;
- $H(c)$ is the normalized entropy of the class attribute, being a measure of the imbalance of the dataset concerning the proportions of observations per class;
- $\bar{M}_{CX}$ is the mean mutual information of the input features and the class and measures the average shared information between the class and the input features;
- DN_{ER} is the error rate of a decision node (aka decision stump), measured in a ten-fold cross-validation procedure;
- $SD(v)$ is the standard deviation of the weighted distance, measuring the sparsity of the observations in a dataset;
- $F3$ is the maximum feature efficiency, measuring whether there is at least an input feature allowing a linear separation of the classes;
- $F4$ is the collective feature efficiency, measuring whether an iterative combination of the input features can separate the classes effectively;
- $L2$ is the training error of a linear classifier and assesses whether the dataset is linearly separable or not;
- $N1$ is the fraction of points on the class boundary, estimating the size of the decision boundary needed to separate the classes by regarding on nearby observations of different classes;
- $N4$ is the nonlinearity of the nearest neighbor classifier, which estimates the nonlinearity of the class boundary needed to separate the classes by measuring the error rate of a one-nearest neighbor classifier for new observations generated by random linear interpolation of some original training observations.

From the list of meta-features, the datasets are placed in the IS regarding: the efficacy of the available input features in separating the classes ($H(X)_{max}$, $\bar{M}_{CX}$,

$F3$ and $F4$), the expected format of the decision frontier ($L2$, $N1$ and $N4$), the imbalance of the classes ($H(c)$), and the sparsity of the data ($SD(v)$).

3.2 IS of regression problems

The set $\mathcal{I}$ of the IS of regression problems has 4885 datasets, where: 246 datasets were collected from the Kell (Alcalá-Fdez et al. 2011), OpenML (Vanschoren et al. 2014), and UCI Machine Learning (Dua and Graff 2017) public repositories; 2547 datasets were randomly selected instances from the Comparing COntinuous Optimisation (COCO) benchmark set (Hansen et al. 2014), commonly used to test numerical optimisation algorithms; 1763 datasets are instances generated for testing black-box optimisation algorithms as described in Muñoz and Smith-Miles (2019); and 299 datasets correspond to time series problems from the M3-Competition (LLC 2019), transformed into regression datasets by using the auto-regressive method, in which previous values of the series are used to predict the next. The selected datasets have from 13 to 2400 observations and from 1 to 108 input features.

The set $\mathcal{A}$ contains the algorithms: Adaboost (ADB), Bagging (BAG), Bayesian ARD (B-ARD), Decision Tree (DT), Support Vector Regressor (ϵ-SVR), linear SVR (l-SVR), ν-SVR, Extra Tree (ET), Gradient Boosting (GB), Kernel Ridge regression (KR), Multilayer Perceptron Neural Network (MLP), Passive aggressive (PA), Random Forest (RF), and Stochastic Gradient Descent (SGD). They belong to different families of algorithms and present distinct biases. Their performance $\mathcal{Y}$ was assessed using a five-fold cross-validation strategy with the Normalized Mean Absolute Error (NMAE) metric, represented in Eq. 12.

$$NMAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{\sum_{i=1}^{n} |y_i - \tilde{y}|} \tag{12}$$

where y_i is the target output for the i-th observation of the dataset, $\hat{y}_i$ is the prediction obtained for this observation, and $\tilde{y}$ is the average of the target values in the dataset.

A set of 26 meta-features were employed to describe the datasets and compose $\mathcal{F}$. After meta-feature selection, seven of them are kept. The meta-dataset $\mathcal{M}$ subject to PBLDR has 4,885 rows (instances or datasets) and 21 columns (seven meta-features plus 14 algorithmic performances). The projection of the datasets in the IS space is given by:

$$\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} = \begin{bmatrix} 0.5655 & 0.5303 \\ -0.1581 & 0.4419 \\ -0.3881 & -0.1159 \\ 0.318 & -0.253 \\ -0.4138 & -0.0987 \\ 0.2884 & -0.4519 \\ 0.092 & 0.2102 \end{bmatrix}^T \begin{bmatrix} n1 \\ C2 \\ C5 \\ L1_a \\ M5 \\ S1 \\ T2 \end{bmatrix} \tag{13}$$

where:

- $n1$ is the number of observations the dataset has or its size;
- $C2$ is the average input features' correlation to the output, measuring whether the input features are predictive of the data labels, considering a linear relationship;
- $C5$ is the average correlation between the input features of the dataset, measuring the degree of redundancy of the input features;
- $L1_a$ is the mean absolute error of and Ordinary Least Square regressor (OLS) without using symbolic features as input, measuring whether a linear fit suits the dataset or not;
- $M5$ is the average mutual information among input features, also quantifying their level of redundancy;
- $S1$ is the normalized output distribution, measuring the smoothness of the relationship of similar observations in the dataset;
- $T2$ is the ratio between the number of observations and the number of input features, being a rough indicator of data sparsity.

Therefore, the meta-features for regression problems used in the IS account for: the representativeness of the available input features ($C2$, $C5$ and $M5$), the size of the dataset ($n1$ and $T2$), the expected format of the approximation function fitting the data ($L1_a$ and $S1$), and data sparsity ($T2$).

3.3 Optimization problem

The proposed methodology focuses on finding $M < N$ subsets S within all datasets $\mathcal{I}$ that have the greatest coverage of all points (datasets) and are equally spaced from each other when projected in the IS. In other words, it seeks to divide the IS into M circular regions where each one has a central dataset that best represents this region. Summarizing, the aim is to find a set of M datasets that represent the entire IS space with greater heterogeneity. To solve this combinatorial optimization problem, an objective function based on the maximum coverage and circle packing problems is proposed. The LA meta-heuristic is then used to solve the formulated optimization problem.

The sets S were defined as circular areas. Having the number of regions or circles to be formed equal to M, the objective is to position these circles in points of the IS that lead to the greatest coverage. In order to give the same priority to each dataset, they must have circles of the same radius R. However, the proposed methodology allows varying the number of datasets to be selected based on a target number that the operator wishes to choose. Therefore, a definition of which should be the radius R as a function of M is needed.

The CP in a square (as is the case of the IS conformation) is a classical optimization problem in the literature that already has exact results for calculating R in a unit square, and therefore expandable to any size, for M up to more than 100 circles. These results allow to say which should be the largest possible radius for these circles that results in the greatest area coverage in the IS square (R_{opt}) (Flores et al. 2016). To reduce the dimensionality of the optimization problem here, a vector that associates M to R_{opt} is incorporated in the optimization problem.

Having defined the size of the circles as a function of the number of datasets to be chosen and knowing that the IS is square, the MC problem arises as where to position these circles in the IS so that this results in the greatest coverage. No circle is allowed to overlap neither to fall outside this square. Still, these circles cannot be positioned anywhere, but at some point that is already contained in the IS itself, that is, an existing dataset.

Two optimization options were considered: discrete or continuous. If discrete, a binary vector of size N would be used, where M positions would get a value 1 (representing the selection of a dataset) and the corresponding datasets could get a circle, making it possible to compute the coverage. However, this formulation would result in $D = 235$ for classification problems and $D = 4885$ for regression problems, where D is the size of the search space, which would be a high-dimensional optimization problem. In the continuous optimization formulation, each dataset is associated with a Cartesian coordinate $Z_M = (z_{1M}, z_{2M})$ and therefore, the dimensionality D of the search space is reduced to $D = 2 * M$. This was the option chosen and in order to circumvent the problem of discrete selection of datasets in the ISA, the LA was allowed to search the entire space and for each proposed solution Z_M, the dataset closest to it in the IS was considered for positioning the circle, as calculated by the Euclidean distance.

Then, for each set of proposed solutions S with M circles (or datasets) of optimal radius (R_{opt}), the coverage is calculated. The considered optimization problem can be formulated as a linear integer programming problem (Matt et al. 2022). To do so, a large number of M' points X_j are randomly plotted across the search space (or IS) and all of them are compared with each Z_M solution using R_{opt}. The more of these M' points are contained within each region of each Z point with radius R_{opt}, the greater the coverage. A constraint is also added to the program that penalizes solutions that repeat datasets.

Therefore, the optimization problem to be solved by LA is an adaptation of the Eq. 5 and can be expressed by Eq. 14. Note that $\min(Z)$ and $\max(Z)$ are the lower and upper bounds composed by the minimum and maximum values found in the IS that forms a square (minimum and maximum found in both z_1 and z_2 IS's axes).

The LA is applied to solve the optimization problem described by Eq. 14 for both classification and regression problems, whose data were obtained from MATILDA.

$$\max \sum_{i=1}^{M'} \sum_{j=1}^{M} y_{ij} \tag{14}$$

Subject to:

$$\min(Z) \leqslant Z_i \leqslant \max(Z) \tag{15}$$

$$d_{i,j} = \sqrt{(Z_i - X_j)^2} \tag{16}$$

$$y_{ij} = 1 \quad \text{if} \quad d_{i,j} \leqslant R_{opt} \tag{17}$$

$$Z_i \neq Z_{i+1} \tag{18}$$

Therefore, the proposed method will be called Lichtenberg-MATILDA (LM) hereafter. In this paper the used LA's parameters are: $R_c = 200$, $N_p = 860{,}000$, $S = 0.88$, $ref = 0.4$, $M = 0$, $Pop = 10 * D$, and $N_{iter} = 100$. They were found after an in-depth study with more than 15,000 simulations using Design of Experiments with Full Factorial design and Response Surface Methodology in 10 complex test functions with more than 10 design variables. A comparative study of LA with three of the most popular meta-heuristics from the literature for the proposed problem are in the Appendix section, where the LA was able to find solutions with better coverage of the IS.

In our formulation, each datataset (or instance) is located according to the corresponding Cartesian coordinates in the two-dimensional IS. For ISs with more dimensions, more decision variables would need to be added, increasing the difficulty and computational cost of the optimization problem. The authors believe that the two-dimensional coordinates from standard ISA (Z_1 and Z_2) are already able to represent all meta-features well, while allowing to present the results graphically, which strengthens the appeal of the applied methodology.

3.4 Baselines

The proposed method is compared with other two selection methods: a simple random selection of M datasets from the pool $\mathcal{I}$ and applying a k-medoids clustering algorithm (Park and Jun 2009) to the IS (where $k = M$). The last technique is a very effective and fast clustering technique able to partition a set of points into k clusters represented by their medoids, using the Eucledian distance for computing the dissimilarity of the pairs of points in the clustered space (Arora and Varshney 2016).

We perform two types of analyses. The first is visual, where the chosen datasets are highlighted in the IS. The second shows plots of the algorithmic performances of the pool of algorithms tested for the datasets. The greater the variation in the performances, the greater the diversity of the datasets concerning their ability to challenge the ML algorithms at different levels (that is, including easy, medium and hard level datasets). Similar plots for the meta-features values are also presented, in order to evidence the diversity regarding properties of the datasets selected.

4 Results and discussion

For both classification and regression IS problems, the following setups will be considered: (i) Finding a set of more diverse datasets that cover the entire IS of Fig. 2; (ii) Repeating the previous step for the quadrants of the IS containing the datasets hardest to predict, which are considered of greater difficulty; Finally, (iii) comparing the proposed optimization method with the other two baseline dataset selection methods considering the entire IS.

4.1 Benchmarks for classification

4.1.1 More diverse datasets

Considering the entire IS, the selection of diversified datasets is guaranteed. The visual results for $M = \{3, 4, 5, 10, 20, 30\}$ datasets are in Fig. 4, where the selected datasets are represented as red dots. It is possible to note that in all cases the proposed methodology fulfilled its goal of selecting datasets that, according to their number M and consequently R_{opt}, covered the entire IS, avoiding as much as possible the overlapping of the selected regions. Furthermore, there was no repetition of datasets for any case.

As the number of selected datasets increases, the method seeks to select datasets near the border of the IS, ensuring a more extreme distance selection. This is clearer from the selection of three datasets (Fig. 4a). In Fig. 4b it is possible to see a complete division, where each quadrant of the IS has a dataset representing it. Up to 20 datasets (Fig. 4e), the total separation of regions is still clear. Due to the greater complexity of the optimization problem for 30 datasets, it is possible to observe in Fig. 4f that some regions begin to slightly overlap.

The names of the selected datasets are presented in Table 1 of the Appendix section. The Lichtenberg-MATILDA (LM) benchmark datasets are quite diverse, although there are some repeated datasets for different M sizes. The most repeated dataset was *teaching*, which was selected in the benchmarks of 3, 4, 5, 20 and 30

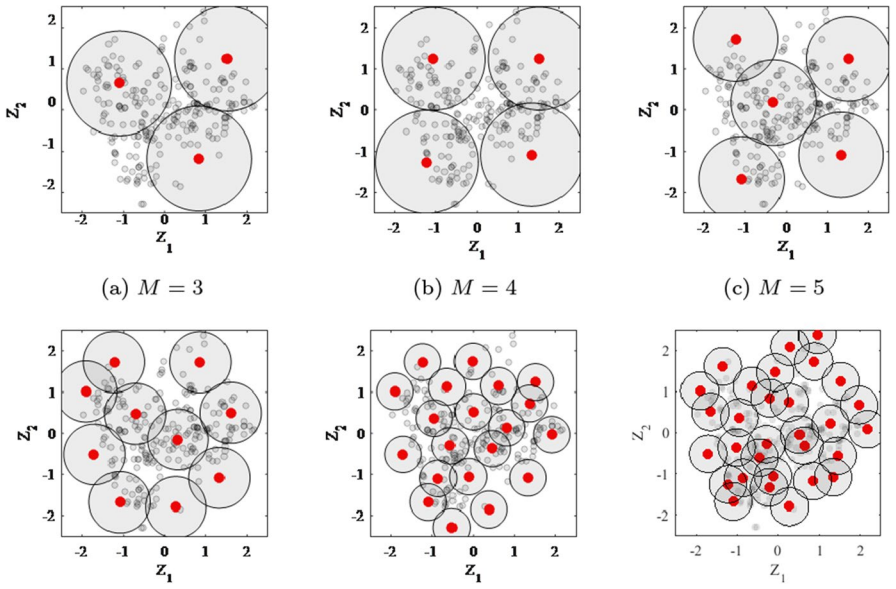


Fig. 4 Selection of more diverse classification datasets: datasets selected by the LM algorithm for different M values are colored in red (Color figure online)

datasets. Next comes the dataset *thyroid_allhyper*, being in benchmarks of 5, 10, 20, and 30 datasets. Considering that the selection here is geometric, the repetition of datasets is justified by their positions in isolated regions with a lower density of points, reinforcing their special properties that are not captured by other datasets.

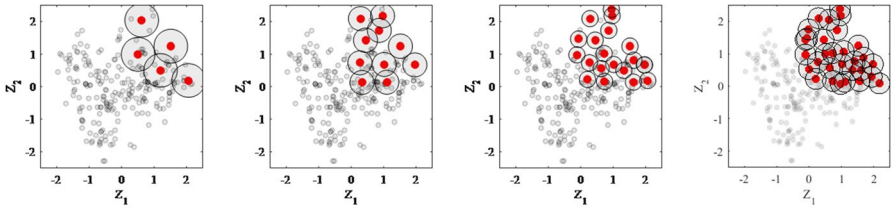
Comparing this study with the only two others that proposed classification benchmark datasets considering more general aspects (Bischl et al. 2017; Olson et al. 2017), some differences are noticeable. The *teaching* dataset, that appeared most in the different selections and is considered of greater difficulty for being in the positive IS quadrant (see Fig. 4a), is not present in the previous benchmarks. The only common datasets with these studies are: (i) *phishing*, *diabetic*, *balance*, and *credit* for Bischl et al. (2017) and (ii) *titanic*, *flare*, *diabetics*, *coil2000*, *mushroom*, *credit*, *automobile*, *yeast*, *crx*, *dermatology*, and *spambase* for Olson et al. (2017). As can be seen, there are few datasets common to all studies, although there are more similarities of our selections with (Olson et al. 2017), who also used MtL to analyze the datasets. But neither of the previous work used any kind of optimization in this selection process, differing from our proposal.

4.2 Hardest datasets

While in the previous analysis the most diverse datasets were considered by adopting all the 235 datasets in the IS for classification, here the same methodology was applied to determine benchmarks with $M = \{5, 10, 20, 30\}$ datasets that are able to challenge the most the ML classification algorithms. These datasets are contained in the IS Quadrant 1 (with $Z_1 \geq 0$ and $Z_2 \geq 0$). The visual results are in Fig. 5 and the datasets names are in Table 2 (appendix section).

Here the IS was reduced by a quarter and the proposed method was automatically adapted by reducing R_{opt}. An increased number of datasets leads to more overlapping of regions, and the LM method really forces the selection of more extreme and distant datasets. This is because the dataset selection does not depend only on their positions in the IS, but on the coverage of the entire search space considered, which makes the proposed method less susceptible to regions of high densities in the space.

In order to show how the difficulty level of the selected datasets increased here compared to selecting datasets from all the IS, Fig. 6 shows boxplots of the error rates (ER as measured by Eq. 10) of the classification techniques composing the pool $\mathcal{A}$. Boxplots in green correspond to the ER values registered for the diverse benchmark of datasets, while boxplots in blue contain the ER registered for the hard datasets. In these cases benchmarks of $M = 30$ datasets are considered. In red are boxplots of the ERs for all 235 datasets in the pool $\mathcal{I}$. Clearly, the error rates are higher for the hard datasets, for all classification techniques considered. On the other hand, the interquartile ranges (IQR) are usually larger for the diverse benchmark, demonstrating it encompasses datasets with distinct hardness profiles (from easy to hard to classify). The boxplots for diverse datasets are in general similar and sometimes larger than those of using all datasets, showing how the proposed methodology successfully captured the entire IS representation in terms of classification difficulty, but with a smaller number of core instances.



(a) $M = 5$ (b) $M = 10$ (c) $M = 20$ (d) $M = 30$

Fig. 5 Hardest classification datasets selected by the LM algorithm, represented as red dots (Color figure online)

For stress-testing how changing the pool of datasets may affect the conclusions drawn from comparing different ML techniques, Fig. 7 shows the Critical Difference (CD) diagrams after comparing the pool of algorithms in $\mathcal{A}$ using the diverse and the hard benchmarks containing $M = 30$ datasets. The Friedman multiple comparison test is employed, followed by the Nemenyi test at 95% of confidence level, as described in Demsar (2006) and Calvo and Santafé Rodrigo (2016). The linear SVM classifier (LSVM) performed better for the diverse benchmark, in general, whilst RF was the best classifier overall in the hard benchmark of datasets. QDA was the worst performing algorithm in both sets. In addition to this non-parametric test, a detailed discussing using the Bayesian test from Benavoli et al. (2017) is in the Appendix section.

The *teaching* dataset is again the most selected in the hard selection, being in all benchmark datasets (of all sizes). Followed by *auto7_2* (benchmarks of 10, 20, and 30 datasets), *heartswitzerland_no_Nas* (10, 20, and 30 datasets), *auto6_3* (10, 20, and 30 datasets) and *hayes* (10, 20, and 30 datasets). The decrease both in the R_{opt} and the number of candidate datasets results in a greater repetition of datasets among benchmarks of different sizes, as expected.

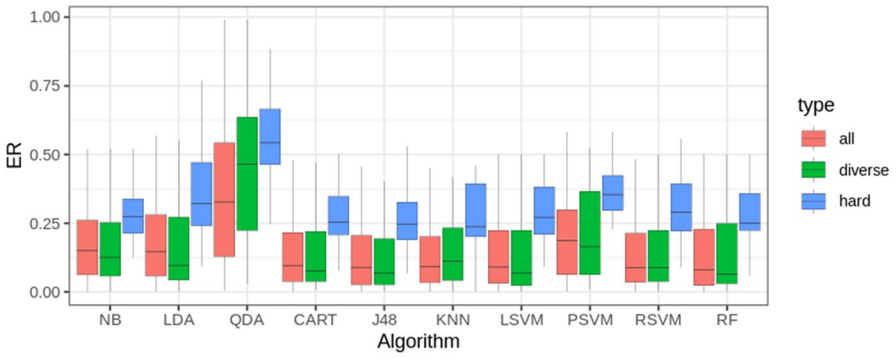
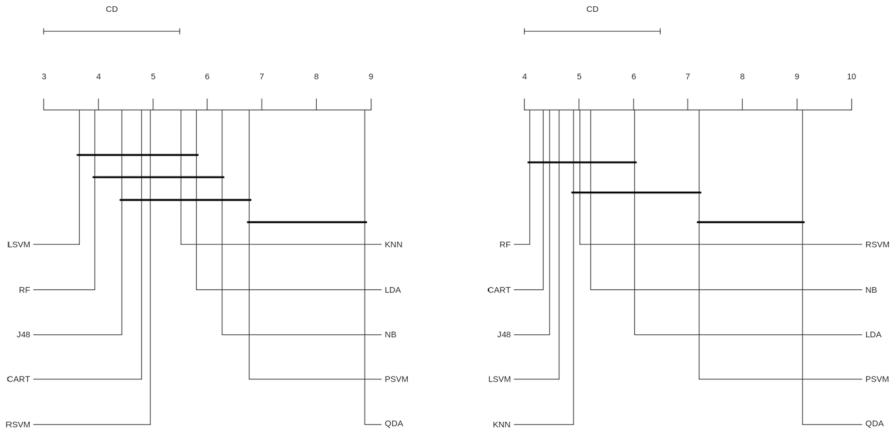


Fig. 6 Boxplots of error rates (ER) of classification techniques in the pool $\mathcal{A}$ for the datasets composing the set of diverse datasets (in green), the set of hard datasets (in blue), for $M = 30$, and all the pool $\mathcal{I}$ of datasets (in red) (Color figure online)



(a) Diverse datasets. (b) Hard datasets.

Fig. 7 Critical Difference diagram of statistical comparison of classifiers using different sets of $M = 30$ datasets (diverse and hard)

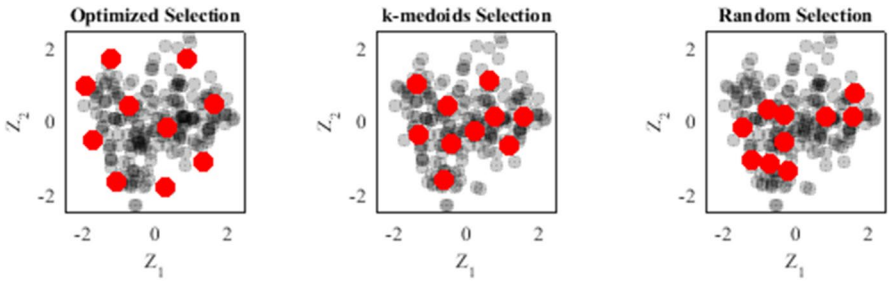
Comparing these selections once again with the benchmark datasets from previous work from the literature, the following are common: (i) *breast*, *dresses*, *madelon*, *credit_no_Nas*, and *cilinder* for Bischl et al. (2017) and (ii) *horse_colic_outcome*, *lymphography*, *breast*, *yeast*, *contraceptive*, *tae*, and *hv_noise* for Olson et al. (2017). Once again, there are few common datasets with LM's selections. Although this suggests these benchmarks have a reduced number of difficult datasets, this might not be the case since they contain other datasets that are absent from our initial pool of instances $\mathcal{I}$.

4.2.1 LM's comparison to baselines

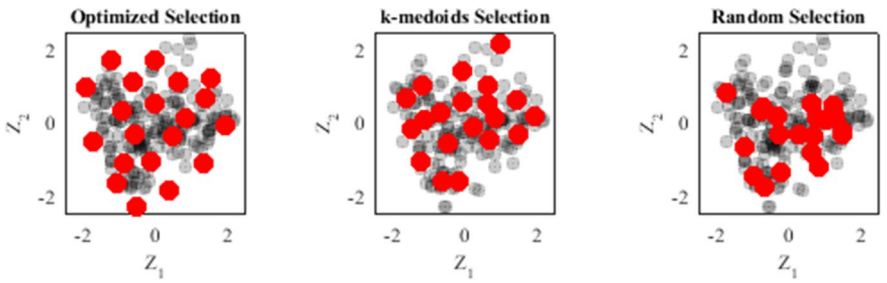
Figures 4 and 5 evidence that the LM optimization method is able to cover well the regions defined in the IS for classification problems. Now the attention is dedicated to compare the obtained selections with those of two baselines: a simple random selection and using the k-medoids clustering technique in the IS with $k = M$.

The visual results of these three methods are shown in Fig. 8, reminding the reader that in all cases the optimized selected datasets are the same ones presented previously, provided by LM. In terms of greater coverage of all the 235 IS datasets, it is clear that the LM method is able to select the most diverse datasets for whatever the number of datasets considered. Considering the benchmarks of 10 datasets (Fig. 8a), LM increased better the distances between the datasets, selecting the most extreme ones. The k-medoids clustering solutions are more concentrated in the central region of the IS, despite having a better distribution of the selected datasets compared to the random selection. The same behavior is repeated for more datasets, with a clear trend that with this increase, the two baseline techniques present difficulties in selecting more distant datasets.

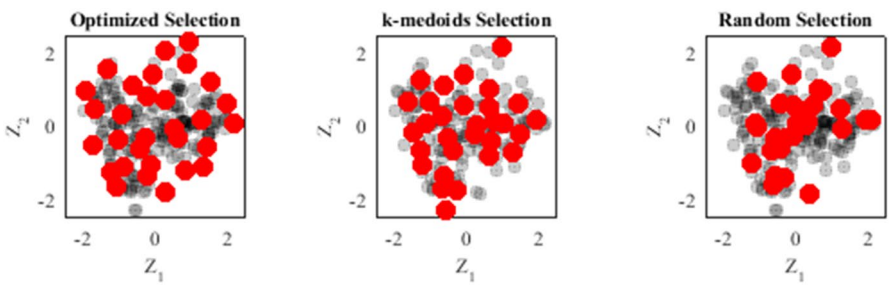
This reinforces what has already been discussed before: the selection of datasets proposed here is not based only on the datasets as plotted in the IS themselves, but on covering the entire IS region. The *k*-medoids clustering technique is based purely on the organization of the points in the IS, which makes its selection result in more internal datasets, less distant from each other, and a more confusing selection in regions of greater density. All these challenges are overcome by the proposed method. However, it is important to emphasize that its computational cost increases linearly with M.



(a) $M = 10$



(b) $M = 20$



(c) $M = 30$

Fig. 8 Classification datasets selected by methods: optimized, k-medoids, and random selection (Color figure online)

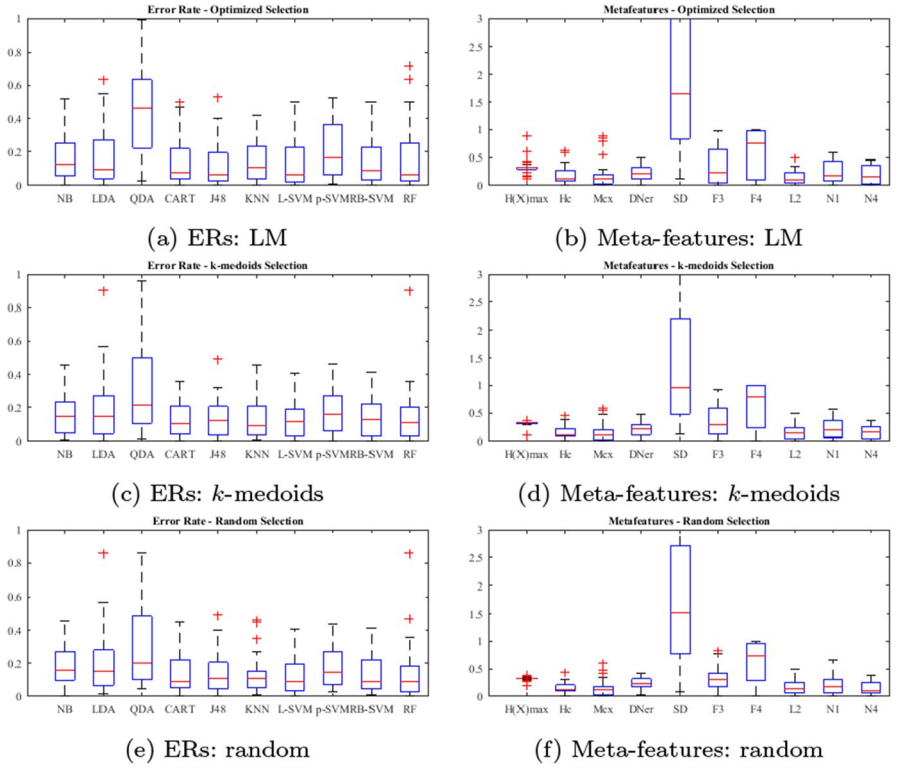


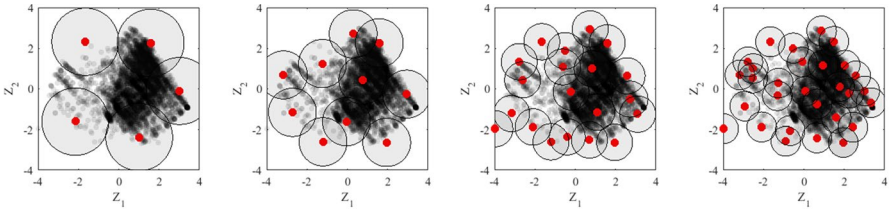
Fig. 9 Error rate of ML classification algorithms (left side) and meta-features values (right side) for selections of $M = 30$ datasets (Color figure online)

In addition to the visual inspection of the datasets selected by each technique, boxplots of the ER performance of each of the algorithms from the pool $\mathcal{A}$ used to build the IS for the datasets selected are shown, for the three benchmark selection methods. The greater the range of the plots, the greater the variety of difficulty levels of the selected datasets. Results for $M = 30$ benchmark datasets are in the left column of Fig. 9, as with 30 datasets all methods are given higher opportunities to select diverse datasets. When presented to the datasets selected by LM, all ML techniques showed a larger variation of ER results, with more elongated boxplots. k-medoids and random selection had in most of the cases similar variations. The same types of observations hold for the meta-features values (Fig. 9b, d and f in the right side), which tend to vary in a larger extent for the datasets selected by LM.

4.3 Benchmarks for regression

4.3.1 More diverse datasets

The IS here has 4885 candidate datasets with many regions of high density (as can be seen in Fig. 2b), which makes the benchmark selection optimization problem



(a) $M = 5$. (b) $M = 10$. (c) $M = 20$. (d) $M = 30$.

Fig. 10 Selection of more diverse regression datasets: datasets selected by the LM algorithm for different M values are colored in red (Color figure online)

more challenging. The visual results of selecting $M = \{5, 10, 20, 30\}$ regression datasets over the entire IS are in Fig. 10, while the datasets names are listed in Table 3 (appendix section).

In contrast to what was observed for classification problems, even for a low number of datasets, there is a slight overlapping of regions, which becomes more accentuated with the increase in the number of datasets selected. This is due to the greater complexity of the search problem here. Still a good spacing between the selected datasets was observed in all cases, with the exception of the upper left quadrant of the benchmark of 30 datasets.

This study is the first, to the authors' best knowledge, to propose a benchmark of datasets for evaluating ML regression algorithms, so there are no previous studies to compare if the datasets found in Table 3 have been suggested before. Due to the high quantity of available datasets in the set $\mathcal{I}$, there was practically no repetition of datasets for benchmarks of different sizes. Some notable exceptions are: *treasury* (appearing in benchmarks of 5, 20 and 30 datasets) and *Data_X_D2_Y_EXP1_82_50_truncated* (composing benchmarks of 10, 20, and 30 datasets).

4.3.2 Hardest datasets

Here the LM algorithm is employed to select benchmarks with $M = \{5, 10, 20, 30\}$ datasets that are able to challenge the ML regression algorithms the most. These datasets are contained in IS Quadrant 3 (with coordinates $Z_1 \leq 0$ and $Z_2 \leq 0$ in Fig. 2b). As this quadrant is less dense compared to the others, for up to 10 datasets there is almost no overlapping of regions found by LM (Fig. 11). However, precisely due to the lack of datasets, the overlap increases after this number. This fact also contributes to a repetition of datasets between benchmarks of different sizes. The names of the selected datasets are in Table 4 (appendix section), where the most repeated datasets for different benchmark sizes are: *N 138_20_1_2_1* (appearing in all benchmarks); *N 35_20_1_2_1* (5, 20, and 30 benchmarks); *dataset_2197_longley* (5, 10, and 20 benchmarks); and *N1167_24_1_2_1* (10, 20, and 30 benchmarks).

The Appendix section show a similar statistical comparison of the regressors for the diverse and hard benchmarks of $M = 30$ datasets too. There are more noticeable differences in the rankings of algorithms for the regression algorithms. For instance, the algorithm Bayesian ARD, which is the best performing

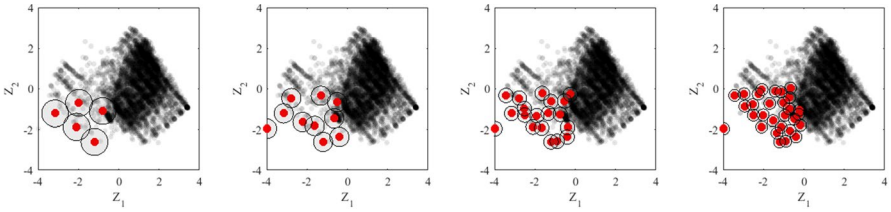


Fig. 11 Hardest regression datasets selected by the LM algorithm, represented as red dots (Color figure online)

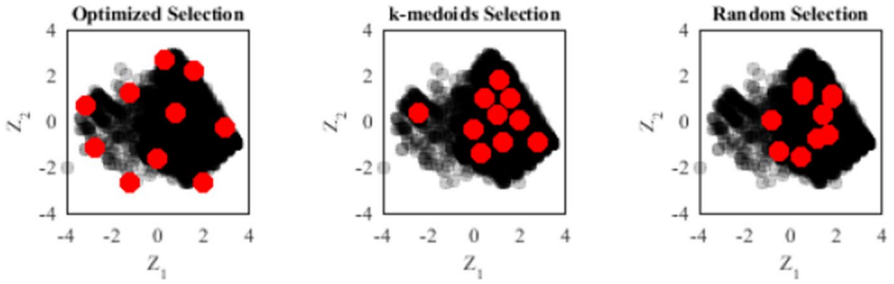
algorithm for the set of hard datasets, is one intermediary solution in the diverse benchmark of datasets. Results of a Bayesian statistical test are also presented in the Appendix section.

4.3.3 LM's comparison to baselines

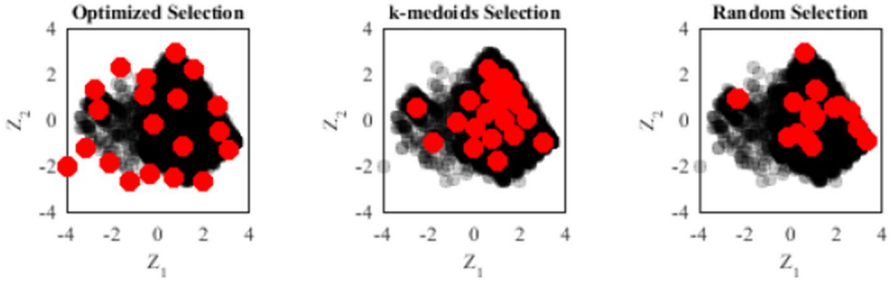
In face of the large number of datasets composing the instance set $\mathcal{I}$ for regression, all selection techniques will be more challenged when choosing a reduced subset. Figure 12 shows how LM, k-medoids, and random selections perform for $M = \{10, 20, 30\}$ datasets.

LM is clearly able to cover the IS better and proves to be even more powerful for a large and dense number of datasets. Again, the optimized method selects datasets that are placed more at the extremes of the IS, making it possible to select datasets with very distinct properties not covered by other denser regions of the space. The k-medoids technique keeps selecting more grouped datasets and concentrated in the central region of the IS, although they are more spaced than the randomly selected datasets. This pattern holds regardless of the number of datasets to be selected and the superiority of the LM method grows with the increase in the number of datasets.

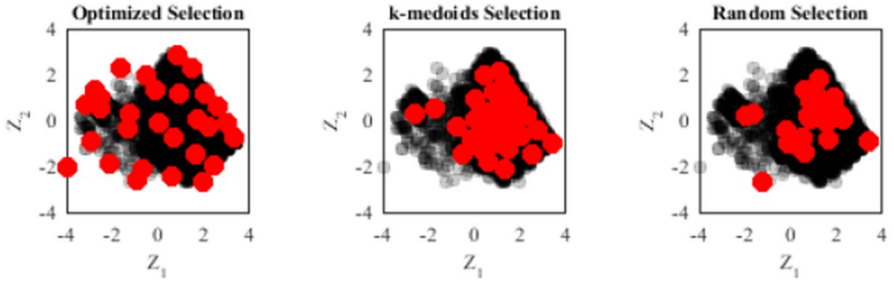
The greater diversity and coverage of the LM technique is also observed by examining the boxplots of the NMAE errors of the regression algorithms used to build the IS in the datasets selected by the three techniques and of their meta-features values. Boxplots of these results for the benchmark of 30 datasets are in Fig. 13. For all three methods, all ML regression algorithms, and all of the meta-features values excepting n_1, the LM technique shows a much larger variation of results. Furthermore, in all cases the mean NMAE for the datasets selected by the LM method are higher. The main reason for this is that the compared methods depend exclusively on the positions of the datasets and their density in the IS. As the most difficult datasets for regression are positioned in the negative quadrant of the IS, which presents a small density, the LM technique is the only technique able to select datasets that best represent this area.



(a) $M = 10$.



(b) $M = 20$.



(c) $M = 30$.

Fig. 12 Regression datasets selected by methods: optimized, k-medoids, and random selection (Color figure online)

5 Conclusion

This work is dedicated to propose and evaluate a method for selecting an unbiased subset of dataset benchmarks that are diverse and challenging in order to evaluate both classification and regression ML algorithms. Leveraging from a 2-D mapping built using meta-features extracted from datasets contained in public repositories, named Instance Space, a complex NP-hard problem is formulated combining Maximum Coverage and Circular Packing and is solved by

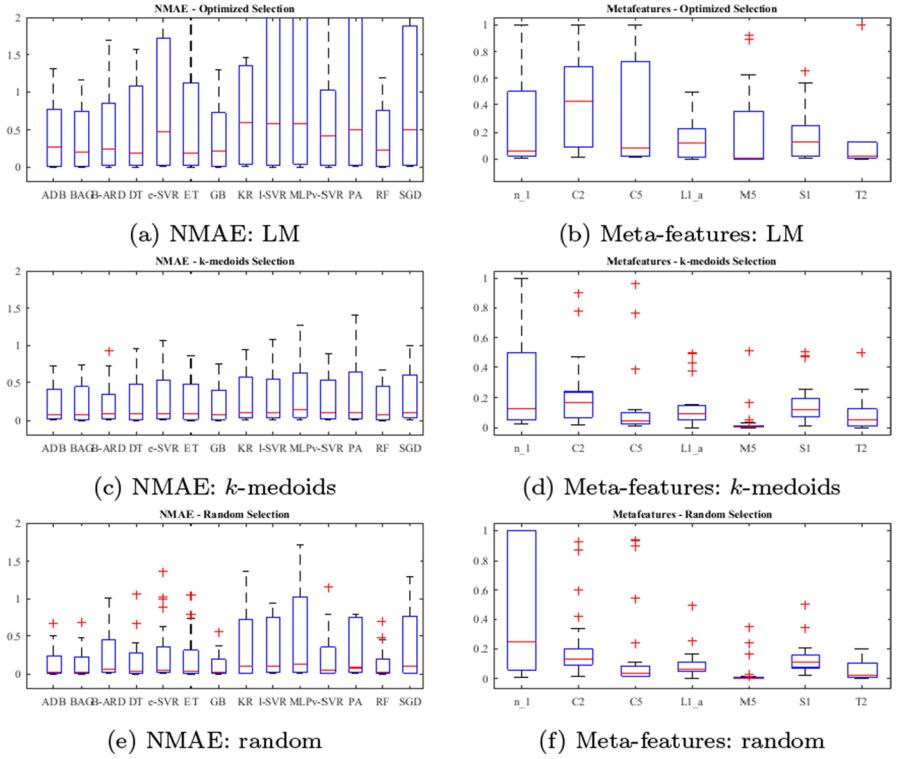


Fig. 13 NMAE of ML regression algorithms (left side) and meta-features values (right side) for selections of $M = 30$ datasets (Color figure online)

the Lichtenberg Algorithm meta-heuristic. The results of the proposed method, named Lichtenberg-MATILDA (LM), are then compared with selections performed by k-medoids clustering and a random selection.

Benchmarks of 5, 10, 20, and 30 datasets containing more diverse and hardest datasets are proposed as case studies, totalling 8 benchmark dataset suites. Applying the proposed method, there was little overlapping of regions covered by different suites in the IS, which grew with the increase in the number of datasets to be selected. Some of the classification datasets found are consistent with previous studies in the literature and for regression, this is the first work to propose a guided benchmark selection. The more diverse optimized benchmark datasets are then compared with the other two baselines. In all cases, the LM technique selected the datasets with the greatest coverage of the IS, which were both the most extreme and the most widely spaced. The *k*-medoids clustering technique selected datasets that were also well spaced, but with a smaller distance between each other in the IS and which are more concentrated in the central regions of the IS. Both LM and *k*-medoids performed better than the random selection. The superiority of the proposed technique became more evident as the number of datasets to be selected and the amount of available datasets increased. The main

reason for the success of the proposed method is that it ensures the entire IS space coverage. This favors that datasets placed in more extreme and less concentrated regions of the IS, and therefore with more distinct characteristics, are selected.

In addition to the IS visual results, the errors of the algorithms used to build the IS in the selected benchmarks are compared, as well as the meta-features values of the datasets. The benchmark datasets selected by LM in general presented the largest ranges of variation, allowing to stress-test different domains where each algorithm performs better or struggles to solve. Still, it was shown that the proposed benchmark with 30 classification datasets had the same diversity as the whole IS.

Therefore, the proposed methodology proved to be quite efficient in ensuring the selection of the most divergent datasets that cover the IS. Future studies shall consider selecting benchmarks of larger sizes and diversifying more the instances contained in the base ISs by including more datasets. And the same reasoning employed here can be easily extended to guide the selection of benchmarks for other learning and optimization problems. Another future work includes measuring the degree of stringency of a benchmark of datasets from a published work. This can be done by projecting the datasets into the IS according to their meta-features values and measuring their degree of IS coverage.

Finally, although we focused our analysis on diverse and hardest datasets, other selections are possible, by simply restricting the search to other quadrants of the IS or by omitting datasets which do not obey a desired constraint. For example, if one wants to test a regression technique only on datasets of large size, the search should concentrate on the upper right quadrant of the regression IS. Allowing the user to set some additional target characteristics of interest is also a worthwhile future investigation. For instance, one might need to select only classification datasets with a high imbalance ratio or number of classes, but with most distinct properties as measured by other meta-features values. This type of selection is straightforward in the method, which can be run for a subset of datasets obeying a given restriction.

A Supplementary Material

A.1 Extra figures and tables

Figure 14 summarizes the Lichtenberg Algorithm.

Table 1 presents the list of classification datasets selected by the LM algorithm, for each benchmark size M.

Table 2 shows the list of classification datasets chosen when only the hardest quadrant of the IS is considered as search space.

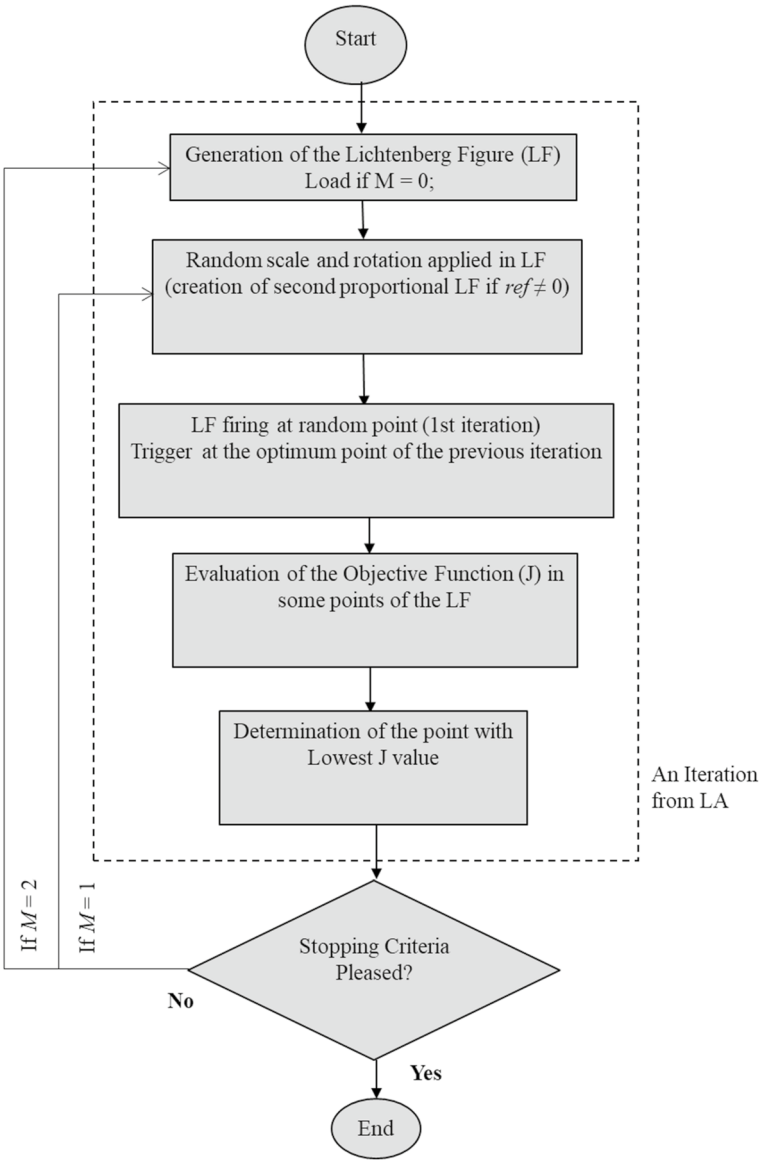


Fig. 14 LA's flowchart

Table 1 The most diverse classification datasets

M	Datasets
2	Titanic and seeds
3	Teaching, seeds, and hiv_schilling
4	Mushroom, teaching, titanic, and audiology_std
5	Flare, thyroid_allhyper, teaching, soybean_small, and titanic
10	chronic_kidney_disease_full_no_Nas, onehund_mar, titanic, seismic, soybean_small, balance, primary, abalone_ori, heart_switzerland_no_Nas, and thyroid_allhyper
20	Qualitative, thyroid_allhyper, credit, soybean_small, thyroid_dis_no_Nas, automobile, heart_va, echocardio, chronic_kidney_disease_full_no_Nas, phishing, titanic, auto7_3, chess-krkp, breast_tissue, onehund_mar, diabetic, congressional, coil2000, teaching, and yeast.
30	breast_tissue, hiv_746, hv_noise, heart_va_no_Nas, congressional, trains, turkiye, horse_colic_lesionm, leaf, phishing, echocardio, seismic, auto8, blood, texture, chronic_kidney_disease_full_no_Nas, user, grammatical_a4, horse_colic_outcome, molecular_splice, auto7_2, robot4, heart_switzerland_no_Nas, onehund_mar, titanic, ozone8_no_Nas, teaching, thyroid_allhyper, mushroom, and hiv_schilling

Table 2 The most challenging classification datasets

M	Datasets
5	Teaching, breast_cancer_wis_pro2, dresses, pitt1_4, and lung_no_NAs
10	Auto7_2, teaching, heart switzerland_no_Nas, trains, horse_colic_outcome, hepatitis_no_Nas, auto6_3, hayes, mechanical, and lung
20	Madelon, hayes, credit_no_Nas, horse_colic_outcome, cilinder_no_NAs, heart_cleverland, auto7_2, heart_va_no_Nas, dresses, lymphography, heart_switzerland_no_Nas, wave2, lung, auto6_3, auto6_2, teaching, trains, auto8, auto7_2, and breast.
30	yeast, lung, auto2, abalone, tae, cilinder, horse_colic_outcome, and auto6_2, trains, contraceptive, madelon, heart_va_no_Nas, teaching, auto8, heart_switzerland_no_Nas, lung_no_NAs, auto7_2, mechanical, automobile, wave2, primary, lenses, hayes, credit_no_Nas, auto6_3, hv_noise, heart_switzerland, heart_cleverland, lymphography, and pitt1_4

Table 3 presents the list of regression datasets selected by the LM algorithm, for each benchmark size *M*.

Table 4 shows the list of regression datasets chosen when only the hardest quadrant of the IS is considered as search space.

Table 3 The most diverse regression datasets

M	Dataset
5	Data_X_D2_Y_EXP1_55_2000_truncated, treasury, N 107_20_1_2_1, Data_X_D2_Y_EXP3_51_2000_truncated, and Data_X_D10_Y_EXP1_82_50_truncated
10	N 81_20_1_2_1, Data_X_D8_Y_F6_2_250, Data_X_D40_Y_F15_6_50, Data_X_D2_Y_EXP1_82_50_truncated, meta, N 138_20_1_2_1, N 391_46_1_5_1, Data_X_D2_Y_EXP1_55_2000_truncated, Data_X_D2_Y_EXP2_16_2000_truncated, and Data_X_D2_Y_F5_22_1000
20	Data_X_D2_Y_EXP2_16_1000_truncated, N 35_20_1_2_1, N1167_24_1_2_1, Data_X_D2_Y_EXP3_12_2000_truncated, treasury, laser, N 861_64_1_6_1, N 138_20_1_2_1, Data_X_D10_Y_EXP2_91_1000_truncated, heart, Data_X_D2_Y_EXP1_55_2000_truncated, N2716_135_1_14_1, detroit, Data_X_D2_Y_EXP2_66_50_truncated Data_X_D3_Y_F5_1_100, Data_X_D2_Y_EXP1_11_2000_truncated, dataset_2197_longley, Data_X_D8_Y_F18_22_500, Data_X_D2_Y_EXP1_82_50_truncated, and Data_X_D5_Y_F12_21_50
30	Data_X_D2_Y_EXP2_7_50_truncated, Data_X_D2_Y_EXP2_47_2000_truncated, Data_X_D2_Y_EXP2_38_250_truncated, Data_X_D2_Y_F12_18_2000, N 391_46_1_5_1, N2716_135_1_14_1, Data_X_D2_Y_EXP3_94_100_truncated, iq_brain_size, N2891_71_1_7_1, Data_X_D2_Y_EXP2_50_2000_truncated, Data_X_D2_Y_EXP3_51_2000_truncated, detroit N 644_36_1_4_1, Data_X_D2_Y_EXP1_82_50_truncated, N 143_20_1_2_1 treasury, Data_X_D2_Y_EXP3_12_2000_truncated, Data_X_D2_Y_EXP2_3_2000_truncated, Data_X_D10_Y_F16_19_1000, Data_X_D2_Y_EXP1_82_50_truncated, vineyard, Data_X_D3_Y_F19_6_50, N 35_20_1_2_1, N2514_120_1_12_1 Data_X_D40_Y_F8_8_1000, Data_X_D10_Y_EXP2_91_250_truncated, Data_X_D20_Y_F5_13_1000, Data_X_D2_Y_EXP1_53_250_truncated N2333_134_1_13_1, Data_X_D10_Y_F16_19_1000, and N1190_24_1_2_1

Table 4 The most challenging regression datasets

Number	Description
5	gascons, N 138_20_1_2_1, N 751_45_1_4_1, N 35_20_1_2_1, and dataset_2197_longley
10	N 138_20_1_2_1, N 95_20_1_2_1, N 456_21_1_2_1, N 229_47_1_5_1, Data_X_D10_Y_EXP3_37_50_truncated, N 576_25_1_2_1, vineyard, N1167_24_1_2_1, detroit, and dataset_2197_longley
20	N 78_20_1_2_1, N 911_72_1_7_1, fri_c3_100_5, N1353_72_1_7_1, N 44_20_1_2_1, N1300_39_1_4_1, N1167_24_1_2_1, dataset_2197_longley, N 35_20_1_2_1, N 102_20_1_2_1, N 523_25_1_2_1, qqdefects_numeric, Data_X_D3_Y_F11_24_50, detroit, N 576_25_1_2_1 N 651_44_1_4_1, N 326_23_1_2_1, qsfsr1, N 138_20_1_2_1, and iq_brain_size
30	N 35_20_1_2_1, N 630_22_1_2_1, N 510_25_1_2_1, N 44_20_1_2_1, Data_X_D100_Y_F19_15_50, N 540_25_1_2_1, N 725_44_1_4_1, N2441_134_1_13_1, qsfsr1, N 518_25_1_2_1, N 823_44_1_4_1, Data_X_D10_Y_F5_13_50, N 138_20_1_2_1, N1167_24_1_2_1, Data_X_D40_Y_F18_12_50, iq_brain_size, N1190_24_1_2_1, N1225_52_1_5_1, N 237_44_1_4_1, N2650_76_1_8_1, qsbr_y2, detroit, auto93, N1327_39_1_4_1, N 78_20_1_2_1, N 143_20_1_2_1, dataset_2191_sleep, N 382_27_1_3_1, N1469_69_1_7_1, analcatdata_neavote

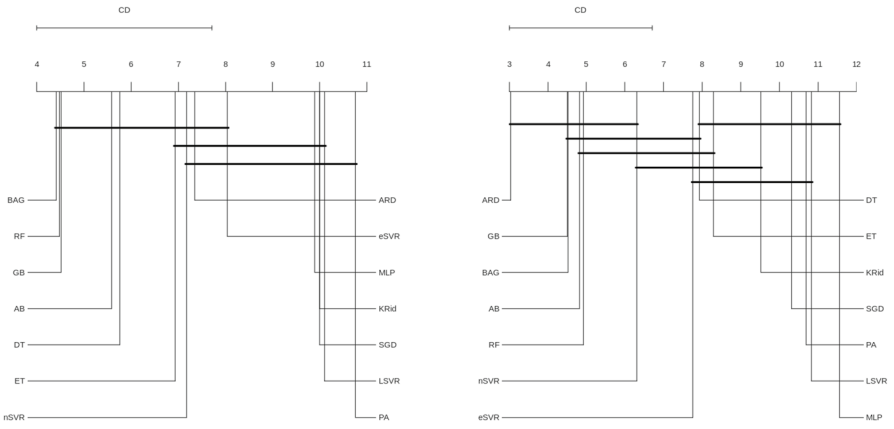
A.2 More non-parametric tests' analysis

Figure 15 shows the CD diagrams after comparing the pool of regression algorithms in $\mathcal{A}$ using the diverse and the hard benchmarks containing $M = 30$ datasets. As in the case of classification problems, the Friedman multiple comparison test is employed, followed by the Nemenyi test at 95% of confidence level (Demsar 2006; Calvo and Santafé Rodrigo 2016). There are more noticeable differences in the rankins of algorithms here. For instance, the algorithm Bayesian ARD (ARD), which is the best performing algorithms for the set of hard datasets, is one of the intermediary solutions in the diverse benchmark of datasets.

In addition to the Friedman multiple comparison non-parametric test used and expressed in Figs. 7 and 15, which is graphically valuable for multiple comparisons along multiple datasets, the Bayesian non-parametric test is also used here. This method allows a more detailed comparison of the performance of the employed algorithms, both regressors and classifiers, in a pairwise comparison (Benavoli et al. 2017; Corani and Benavoli 2015).

Starting from the performance difference between two algorithms in all the datasets, this test calculates the probability p of the algorithm to be the best in these datasets. Or even the probability that both are equivalent, through the determination of a region of practical equivalence (rope). Therefore, three probabilistic regions have to be determined. However, having 10 classifiers and 14 regressors in this work, this would result in 45 and 91 combinations, respectively. Considering diverse and hardest datasets, there would be 272 combinations to apply the test. Since the Friedman-test pointed out the best ranked algorithms in each of the cases, these will be used as reference to be compared with the others.

Table 5 shows the Bayesian test results for the ML algorithms on the diverse and difficult classification datasets. In both, Classifier 1 is the one with the best ranking



(a) Diverse datasets. (b) Hard datasets.

Fig. 15 Critical Difference diagram of statistical comparison of regressors using different sets of $M = 30$ datasets (diverse and hard)

Table 5 Results of Bayesian test between the best ranked classifier in the Friedman test and the other classifiers for the benchmark of 30 datasets

Classif. 2	p (Classif. 1)	p (rope)	p (Classif. 2)
	Diverse datasets (Classif. 1 is LSVM)		
NB	99.8	0.2	0.0
LDA	84.6	15.4	0.0
QDA	100.0	0.0	0.0
CART	77.5	16.0	6.5
J48	46.8	27.0	26.2
KNN	83.30	13.0	3.70
PSVM	99.9	0.1	0.0
RSVM	39.7	60.3	0.0
RF	29.1	37.5	33.4
	Hardest datasets (Classif. 1 is RF)		
NB	69.5	0.0	30.5
LDA	99.1	0.0	0.9
QDA	100.0	0.0	0.0
CART	34.4	0.6	65.0
J48	41.3	0.0	58.7
KNN	62.6	0.0	37.4
LSVM	76.1	2.7	21.2
PSVM	100.0	0.0	0.0
RSVM	93.4	0.5	6.1

in the Friedman test. The results coincide with those of the Friedman test in pointing out that LSVM overcomes the results of NB, PSVM and QDA for diverse datasets, with a large certainty. In the hardest datasets, RF was best ranked and there is a large confidence (higher than 90%) that its results are superior to those of QDA, PSVM, LDA and RSVM. In the Friedman test (diagram of Fig. 7b), the differences between RF, LDA, and RSVM are not conclusive.

Table 6 brings the results of the Bayesian test for the regressors on the diverse and hardest regression datasets. For diverse regression datasets, BAG was the best ranked algorithm and is compared against the other regressors. BAG has outperformed most of the algorithms, with the exception of RF and GB, where the rope probability precludes this assertion. Still, it can be seen that it is only slightly better than ARD. Dealing with hard datasets, there is no doubt that for all datasets the ARD algorithm is the most accurate, whilst in the Friedman test ARD has shown similar performance to GB, BAG, AB, RF and nSVR.

A.3 Comparison to other meta-heuristics

Meta-heuristics are nature-inspired optimization algorithms that computationally assemble some natural behavior to explore and exploit search spaces to find the best

Table 6 Results of Bayesian test between the best ranked regressor in the Friedman test and the other regressors for the benchmard of 30 diverse datasets

Class. 2	p (Class. 1)	p (rope)	p (Class. 2)
	Diverse datasets (Class. 1 is BAG)		
AB	83.9	13.6	2.5
ARD	51.6	0.1	48.3
DT	68.9	10.7	20.4
eSVR	100.0	0.0	0.0
ET	96.3	0.9	2.8
GB	11.9	56.7	31.4
KRid	100.0	0.0	0.0
LSVR	100.0	0.0	0.0
MLP	99.9	0.0	0.1
nSVR	98.9	0.1	1.0
PA	100.0	0.0	0.0
RF	2.9	96.8	0.3
SGD	100.0	0.0	0.0
	Hardest datasets (Class. 1 is B-ARD)		
AB	99.9	0.0	0.1
BAG	100.0	0.0	0.0
DT	100.0	0.0	0.0
eSVR	100.0	0.0	0.0
ET	100.0	0.0	0.0
GB	99.5	0.0	0.5
KRid	100.0	0.0	0.0
LSVR	100.0	0.0	0.0
MLP	100.0	0.0	0.0
nSVR	100.0	0.0	0.0
PA	100.0	0.0	0.0
RF	100.0	0.0	0.0
SGD	100.0	0.0	0.0

possible solutions. They can be divided according to their inspiration creation and basis into the following categories: (i) evolutionary (most common); (ii) swarms; (iii) physical phenomena; and (iv) human behaviors. Besides this, they can be divided according to their search strategies into population and trajectory-based, being the former the category that presents the vast majority of known algorithms (Yang 2020).

In recent years, the literature has brought an explosion of meta-heuristic applications in optimization problems, overlapping with classical and gradient-based methods. Some of the factors that contribute to their success are: (i) better ability to escape from local optima; (ii) better ability to deal with multimodal, convex, and discrete problems; (iii) better capacity to deal with many variables and objectives; (iv) gradient independence; (v) independence from explicit equations, since they can be, for

example, easily associated with numerical analysis software and ML algorithms to give responses from inputs, among others (Kumar et al. 2023; Pereira et al. 2021c).

Also according to the no free-lunch theorem, there is no single meta-heuristic that can be the best in all applications and they compete to deliver the best results at the lowest computational cost (Wolpert 2002; Joyce and Herrmann 2018). As seen before, the optimization problem proposed in this study is combinatorial and was solved in the paper with the LA algorithm. But, for fair comparison, other three meta-heuristics are applied here: GA, PSO, and DE. They are the most popular and classical meta-heuristics and have several good reported results (Yang 2020). All these algorithms have as common parameters the population size and number of iterations.

The GA is the most popular evolution-based meta-heuristic in the literature and is inspired by the natural selection phenomenon and genetics in biology. The agents with best fitness survive and the others tends to vanish. It uses the principles of reproduction, crossover, and mutation to guide the population in the search space through the generations. Crossover improves exploitation, while the mutation guarantees better exploration. Its particular parameters are crossover and mutation rates.

DE has the similar inspiration to GA, but at each iteration it randomly selects three agents in the entire population and combines their characteristics. Its particular parameters are crossover rate (probability that a new solution will be created by the three agents) and differential weight (distance between them).

PSO is the most popular swarm-based optimizer and is inspired by the bird flocking social behavior, where a set of particles (potential solutions) moves around the search space by updating their positions based on their own best position and the best position found by the swarm. It has three particular parameters: cognitive factor (attraction between the particle and its personal best position), social factor

Table 7 Meta-heuristics settings

Meta-heuristic	Parameter	Value
	Common	
All	Population	10*d
	Number of iterations	100
	Particular	
GA	Crossover rate	0.8
	Mutation rate	0.01
PSO	Cognitive factor	2
	Social factor	2
	Inertia weight	0.9
DE	Crossover rate	0.9
	Differential weight	0.5
LA	Creation radius	200
	Particles number	860,000
	Stickness coefficient	0.88
	Switching factor	0
	Refinement	0.4

Table 8 Meta-heuristics results for the selection of 10 datasets

Meta-heuristic	Mean	Standard deviation	Simulations time (s)
DE	0.6155	0.0059	**4877.30**
PSO	0.6370	0.0055	5992.99
GA	0.6434	0.0097	26,161.75
LA	**0.6442**	0.0043	14,244.13

(attraction between the particle and the swarm's best position), and inertia weight (controls the impact of the particle's previous speed on its present speed).

The main parameters and the recommended values by the authors that published these algorithms are in Table 7. Beyond these parameters, the population size and number of iterations are shared between all algorithms. They are set to ten times the number of optimization variables and one hundred, respectively (Yang 2020).

The algorithms with the parameters in Table 7 were applied in the problem of Eq. 14 for the Classification IS to select ten diverse datasets, which results in twenty design variables. The only objective here is to observe which of the four algorithms finds the maximum coverage of the IS. A number pf 10 datasets was chosen because it represents a median dimensionality among those adopted in this study, with a moderate computational cost. Running all cases would be computational costly and comparing metaheuristics is not the main purpose of this study. All simulations were run using the software R2022b MATLAB on a CORE i7 Dell computer with 8GB and 1 TB HHD. Each meta-heuristic was run 10 times. The mean and standard deviation of the maximum coverage result and the total time spent on the simulations are in Table 8.

LA was the most accurate technique, finding the best maximum coverage values on average (in bold in Table 8) for the problem and with a lower standard deviation. Next comes GA, PSO and DE. However, it had the third highest computational cost, behind DE and PSO, respectively. Since the algorithm is run in advance in order to select a benchmark that will be used multiple times, our choice was for the technique with highest accuracy and more stable results in the problem.

Author contributions JLJP implemented the optimization framework dedicated to benchmark selection and has run the computational experiments from the paper. KS-M is the proponent of the ISA framework. MAM has implemented the MATILDA tool and generated the instance spaces for classification and regression problems. ACL proposed and supervised all the work from this paper. All authors contributed with paper writing and organization.

Funding This work was partially supported by the Brazilian research agencies CNPq (Grant 307892/2020-4) and FAPESP (Grants 2021/06870-3 and 2022/10683-7). The Australian authors gratefully acknowledge funding from the Australian Research Council (Grant IC200100009) provided to the ARC Training Centre in Optimisation Technologies, Integrated Methodologies and Applications (OPTIMA).

Availability of data and materials The benchmark datasets and outputs of their analysis are available at (https://matilda.unimelb.edu.au/matilda/).

Code availability The source code for Lichtenberg-MATILDA algorithm is in https://www.mathworks.com/matlabcentral/fileexchange/123930-lichtenberg-algorithm-for-benchmark-datasets-selection.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Aguiar GJ, Santana EJ, de Carvalho AC, Junior SB (2022) Using meta-learning for multi-target regression. Inf Sci 584:665–684

Alcalá-Fdez J, Fernández A, Luengo J, Derrac J, García S, Sánchez L, Herrera F (2011) Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. J Multiple-Valued Logic Soft Comput 17:255–287

Alipour H, Muñoz MA, Smith-Miles K (2023) Enhanced instance space analysis for the maximum flow problem. Eur J Oper Res 304(2):411–428

Arora P, Varshney S et al (2016) Analysis of k-means and k-medoids algorithm for big data. Procedia Comput Sci 78:507–512

Bang-Jensen J, Gutin G, Yeo A (2004) When the greedy algorithm fails. Discret Optim 1(2):121–127

Benavoli A, Corani G, Demšar J, Zaffalon M (2017) Time for a change: a tutorial for comparing multiple classifiers through Bayesian analysis. J Mach Learn Res 18(1):2653–2688

Bischl B, Casalicchio G, Feurer M, Hutter F, Lang M, Mantovani RG, van Rijn JN, Vanschoren J (2017) Openml benchmarking suites. arXiv: Machine Learning

Botchkarev A (2018) Performance metrics (error measures) in machine learning regression, forecasting and prognostics: properties and typology. arXiv preprint arXiv:1809.03006

Broyden CG (1970) The convergence of a class of double-rank minimization algorithms 1. General considerations. IMA J Appl Math 6(1):76–90

Calvo B, Santafé Rodrigo G (2016) scmamp: statistical comparison of multiple algorithms in multiple problems. The R Journal, Vol 8/1, Aug 2016

Castillo I, Kampas FJ, Pintér JD (2008) Solving circle packing problems by global optimization: numerical results and industrial applications. Eur J Oper Res 191(3):786–802

Clement CL, Kauwe SK, Sparks TD (2020) Benchmark aflow data sets for machine learning. Integr Mater Manuf Innov 9(2):153–156

Cohen R, Katzir L (2008) The generalized maximum coverage problem. Inf Process Lett 108(1):15–22

Corani G, Benavoli A (2015) A Bayesian approach for comparing cross-validated algorithms on multiple data sets. Mach Learn 100(2–3):285–304

Davenport TH, Ronanki R (2018) Artificial intelligence for the real world. Harv Bus Rev 96(1):108–116

Demsar J (2006) Statistical comparisons of classifiers over multiple datasets. J Mach Learn Res 7:1–30

Dua D, Graff C (2017) UCI machine learning repository. http://archive.ics.uci.edu/ml

Dueben PD, Schultz MG, Chantry M, Gagne DJ, Hall DM, McGovern A (2022) Challenges and benchmark datasets for machine learning in the atmospheric sciences: definition, status, and outlook. Artif Intell Earth Syst 1(3):e210002

Ferri C, Hernández-Orallo J, Modroiu R (2009) An experimental comparison of performance measures for classification. Pattern Recogn Lett 30(1):27–38

Flores JJ, Martínez J, Calderón F (2016) Evolutionary computation solutions to the circle packing problem. Soft Comput 20(4):1521–1535

Garcia LP, Lorena AC, de Souto M, Ho TK (2018) Classifier recommendation using data complexity measures. In: IEEE Proceedings of ICPR 2018

Hannousse A, Yahiouche S (2021) Towards benchmark datasets for machine learning based website phishing detection: an experimental study. Eng Appl Artif Intell 104:104347

Hansen N, Auger A, Finck S, Ros R (2014) Real-parameter black-box optimization benchmarking BBOB-2010: Experimental setup. Tech. Rep. RR-7215, INRIA, http://coco.lri.fr/downloads/download15.02/bbobdocexperiment.pdf

Hochbaum DS (1996) Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems. In: Approximation algorithms for NP-hard problems, pp 94–143

Hooker JN (1995) Testing heuristics: we have it all wrong. J Heurist 1:33–42

Hu W, Fey M, Zitnik M, Dong Y, Ren H, Liu B, Catasta M, Leskovec J (2020) Open graph benchmark: datasets for machine learning on graphs. Adv Neural Inf Process Syst 33:22118–22133
Janairo AG, Baun JJ, Concepcion R, Relano RJ, Francisco K, Enriquez ML, Bandala A, Vicerra RR, Alipio M, Dadios EP (2022) Optimization of subsurface imaging antenna capacitance through geometry modeling using archimedes, lichtenberg and henry gas solubility metaheuristics. In: 2022 IEEE international IOT, electronics and mechatronics conference (IEMTRONICS), IEEE, pp 1–8
Joyce T, Herrmann JM (2018) A review of no free lunch theorems, and their implications for metaheuristic optimisation. In: Yang XS (ed) Nature-inspired algorithms and applied optimization. Springer, Cham, pp 27–51
Khuller S, Moss A, Naor JS (1999) The budgeted maximum coverage problem. Inf Process Lett 70(1):39–45
Kumar A, Nadeem M, Banka H (2023) Nature inspired optimization algorithms: a comprehensive overview. Evol Syst 14(1):141–156
LLC M (2019) International institution of forecasters. https://forecasters.org/resources/time-series-data/m3-competition/
Lorena AC, Maciel AI, de Miranda PB, Costa IG, Prudêncio RB (2018) Data complexity meta-features for regression problems. Mach Learn 107(1):209–246
Lorena AC, Garcia LP, Lehmann J, Souto MC, Ho TK (2019) How complex is your classification problem? A survey on measuring classification complexity. ACM Comput Surv (CSUR) 52(5):1–34
Luengo J, Herrera F (2015) An automatic extraction method of the domains of competence for learning classifiers using data complexity measures. Knowl Inf Syst 42(1):147–180
Ma BJ, Pereira JLJ, Oliva D, Liu S, Kuo YH (2023) Manta ray foraging optimizer-based image segmentation with a two-strategy enhancement. Knowl Based Syst 28:110247
Macià N, Bernadó-Mansilla E (2014) Towards UCI+: a mindful repository design. Inf Sci 261:237–262
Matt PA, Ziegler R, Brajovic D, Roth M, Huber MF (2022) A nested genetic algorithm for explaining classification data sets with decision rules. arXiv preprint arXiv:2209.07575
Muñoz MA, Smith-Miles KA (2019) Generating new space-filling test instances for continuous black-box optimization. Evolut Comput. https://doi.org/10.1162/evco_a_00262
Muñoz MA, Smith-Miles K (2020) Generating new space-filling test instances for continuous black-box optimization. Evol Comput 28(3):379–404
Munoz MA, Villanova L, Baatar D, Smith-Miles K (2018) Instance spaces for machine learning classification. Mach Learn 107(1):109–147
Muñoz MA, Yan T, Leal MR, Smith-Miles K, Lorena AC, Pappa GL, Rodrigues RM (2021) An instance space analysis of regression problems. ACM Trans Knowl Discov Data (TKDD) 15(2):1–25
Nascimento AI, Bastos-Filho CJ (2010) A particle swarm optimization based approach for the maximum coverage problem in cellular base stations positioning. In: 2010 10th international conference on hybrid intelligent systems, IEEE, pp 91–96
Olson RS, La Cava W, Orzechowski P, Urbanowicz RJ, Moore JH (2017) PMLB: a large benchmark suite for machine learning evaluation and comparison. BioData Min 10(1):1–13
Orriols-Puig A, Macia N, Ho TK (2010) Documentation for the data complexity library in C++. Universitat Ramon Llull La Salle 196(1–40):12
Paleyes A, Urma RG, Lawrence ND (2022) Challenges in deploying machine learning: a survey of case studies. ACM Comput Surv 55(6):1–29
Park HS, Jun CH (2009) A simple and fast algorithm for k-medoids clustering. Expert Syst Appl 36(2):3336–3341
Pereira JLJ, Francisco MB, da Cunha Jr SS, Gomes GF (2021a) A powerful Lichtenberg optimization algorithm: a damage identification case study. Eng Appl Artif Intell 97:104055
Pereira JLJ, Francisco MB, Diniz CA, Oliver GA, Cunha SS Jr, Gomes GF (2021b) Lichtenberg algorithm: a novel hybrid physics-based meta-heuristic for global optimization. Expert Syst Appl 170:114522
Pereira JLJ, Oliver GA, Francisco MB, Cunha SS, Gomes GF (2021c) A review of multi-objective optimization: methods and algorithms in mechanical engineering problems. Arch Comput Methods Eng. https://doi.org/10.1007/s11831-021-09663-x
Pereira JLJ, Francisco MB, de Oliveira LA, Chaves JAS, Cunha SS Jr, Gomes GF (2022a) Multi-objective sensor placement optimization of helicopter rotor blade based on feature selection. Mech Syst Signal Process 180:109466
Pereira JLJ, Francisco MB, Ribeiro RF, Cunha SS, Gomes GF (2022b) Deep multiobjective design optimization of CFRP isogrid tubes using Lichtenberg algorithm. Soft Comput 26:7195–7209

Pereira JLJ, Oliver GA, Francisco MB, Cunha SS Jr, Gomes GF (2022c) Multi-objective Lichtenberg algorithm: a hybrid physics-based meta-heuristic for solving engineering problems. Expert Syst Appl 187:115939
Rahmani O, Naderi B, Mohammadi M, Koupaei MN (2018) A novel genetic algorithm for the maximum coverage problem in the three-level supply chain network. Int J Ind Syst Eng 30(2):219–236
Ristoski P, Vries GKDd, Paulheim H (2016) A collection of benchmark datasets for systematic evaluations of machine learning on the semantic web. In: International semantic web conference. Springer, pp 186–194
Rivolli A, Garcia LP, Soares C, Vanschoren J, de Carvalho AC (2022) Meta-features for meta-learning. Knowl-Based Syst 240:108101
Smith-Miles K, Muñoz MA (2023) Instance space analysis for algorithm testing: methodology and software tools. ACM Comput Surv. https://doi.org/10.1145/3572895
Smith-Miles KA (2009) Cross-disciplinary perspectives on meta-learning for algorithm selection. ACM Comput Surv (CSUR) 41(1):6
Soares C (2009) UCI++: improved support for algorithm selection using datasetoids. In: Pacific-Asia conference on knowledge discovery and data mining. Springer, pp 499–506
Takamoto M, Praditia T, Leiteritz R, MacKinlay D, Alesiani F, Pflüger D, Niepert M (2022) Pdebench: an extensive benchmark for scientific machine learning. arXiv preprint arXiv:2210.07182
Taşdemir A, Demirci S, Aslan S (2022) Performance investigation of immune plasma algorithm on solving wireless sensor deployment problem. In: 2022 9th international conference on electrical and electronics engineering (ICEEE), IEEE, pp 296–300
Thiyagalingam J, Shankar M, Fox G, Hey T (2022) Scientific machine learning benchmarks. Nat Rev Phys 4(6):413–420
Tian Z, Wang J (2022) Variable frequency wind speed trend prediction system based on combined neural network and improved multi-objective optimization algorithm. Energy 254:124249
Tossa F, Abdou W, Ansari K, Ezin EC, Gouton P (2022) Area coverage maximization under connectivity constraint in wireless sensor networks. Sensors 22(5):1712
Vanschoren J (2019) Meta-learning. In: Hutter F, Kotthoff L, Vanschoren J (eds) Automated machine learning. Springer, Cham, pp 35–61
Vanschoren J, Van Rijn JN, Bischl B, Torgo L (2014) Openml: networked science in machine learning. ACM SIGKDD Explor Newsl 15(2):49–60
Witten TA Jr, Sander LM (1981) Diffusion-limited aggregation, a kinetic critical phenomenon. Phys Rev Lett 47(19):1400
Wolpert DH (2002) The supervised learning no-free-lunch theorems. In: Roy R, Koppen M, Ovaska S, Furuhashi T, Hoffmann F (eds) Soft computing and industry. Springer, London, pp 25–42
Xiao H, Cheng Y (2022) The image segmentation of Osmanthus fragrans based on optimization algorithms. In: 2022 4th international conference on advances in computer technology. Information science and communications (CTISC), IEEE, pp 1–5
Yang XS (2020) Nature-inspired optimization algorithms. Academic Press, New York
Yarrow S, Razak KA, Seitz AR, Seriès P (2014) Detecting and quantifying topography in neural maps. PLoS ONE 9(2):e87178
Yuan Y, Tole K, Ni F, He K, Xiong Z, Liu J (2022) Adaptive simulated annealing with greedy search for the circle bin packing problem. Comput Oper Res 144:105826
Zhang Z, Schwartz S, Wagner L, Miller W (2000) A greedy algorithm for aligning DNA sequences. J Comput Biol 7(1–2):203–214

Authors and Affiliations

João Luiz Junho Pereira[1] · Kate Smith-Miles[2] · Mario Andrés Muñoz[3] · Ana Carolina Lorena[1]

✉ João Luiz Junho Pereira
joaoluizjp@gmail.com.br

Kate Smith-Miles
smith-miles@unimelb.edu.au

Mario Andrés Muñoz
munoz.m@unimelb.edu.au

Ana Carolina Lorena
aclorena@ita.br

[1] Instituto Tecnológico de Aeronáutica, São José dos Campos, Brazil

[2] School of Mathematics and Statistics, University of Melbourne, Melbourne, Australia

[3] School of Computer and Information Systems, University of Melbourne, Melbourne, Australia