Hybrid approaches to optimization and machine learning methods: a systematic literature review

Beatriz Flamia Azevedo[1,2,3] · Ana Maria A. C. Rocha[3] · Ana I. Pereira[1,2,3]

© The Author(s) 2024

Abstract

Notably, real problems are increasingly complex and require sophisticated models and algorithms capable of quickly dealing with large data sets and finding optimal solutions. However, there is no perfect method or algorithm; all of them have some limitations that can be mitigated or eliminated by combining the skills of different methodologies. In this way, it is expected to develop hybrid algorithms that can take advantage of the potential and particularities of each method (optimization and machine learning) to integrate methodologies and make them more efficient. This paper presents an extensive systematic and bibliometric literature review on hybrid methods involving optimization and machine learning techniques for clustering and classification. It aims to identify the potential of methods and algorithms to overcome the difficulties of one or both methodologies when combined. After the description of optimization and machine learning methods, a numerical overview of the works published since 1970 is presented. Moreover, an in-depth state-of-art review over the last three years is presented. Furthermore, a SWOT analysis of the ten most cited algorithms of the collected database is performed, investigating the strengths and weaknesses of the pure algorithms and detaching the opportunities and threats that have been explored with hybrid methods. Thus, with this investigation, it was possible to highlight the most notable works and discoveries involving hybrid methods in terms of clustering and classification and also point out the difficulties of the pure methods and algorithms that can be strengthened through the inspirations of other methodologies; they are hybrid methods.

Keywords Machine learning · Optimization · Hybrid methods · Literature review · Clustering · Classification

1 Introduction

Mathematical models are present in almost every area of science, as they play a vital role in problem-solving. These models provide a simplified representation of reality from mathematical formulations. These formulations make it possible to understand

complex systems, solve problems, and obtain essential information to support intelligent decision-making. Mathematical models use algorithms to find the most appropriate solution to the problem described.

Numerical Optimization is a well-known area of Mathematical Sciences that aims to identify extreme points of a function, whether maximum or minimum points. Optimization methods have become a crucial tool for management, decision-making, technology improvement, and development in the last two decades, providing competitive advantages to various systems (Mitchell, 1996). Thus, optimization models and algorithms gained visibility in several areas, such as industry (Fera et al., 2018), disease diagnoses (Agustina et al., 2019), professional and resources scheduling and allocation (Alves et al., 2018; Azevedo et al., 2021), financial area with capital management and scenarios forecast (Li et al., 2019), among others.

Another well-known area focused on solving problems using mathematical models and algorithms is Machine Learning. In some real-world problems, much information (data) should be processed. This amount of data usually requires computational assistance to transform the data information into relevant knowledge for problem solving. In this context, machine learning algorithms are extremely useful. These models and algorithms intend to generate a mathematical model that describes the data set and generalizes the knowledge to unknown data samples (Azevedo et al., 2019; Fürnkranz et al., 2012). Machine learning models and algorithms also have application in several domains named: industry (Azevedo et al., 2019), health (Cherif, 2018), financial (Cicceri et al., 2020), education (Agrusti et al., 2019; Buenaño-Fernandez et al., 2019; Zhu, 2019; Azevedo et al., 2022).

Due to the practical importance of both areas, many algorithms to tackle optimization or machine learning problems have been developed. Although the vast majority of algorithms are efficient in solving the problems they face, no one is completely perfect. Numerous limitations are listed in the literature for both optimization and machine learning algorithms (Karaboga et al., 2020; Mehta et al., 2020; Telikani et al., 2021; Wu, 2019). Nevertheless, in recent times, many researchers have started to identify ways to combine both methodologies to overcome the weaknesses of each method, strengthening procedures inspired by other methodologies; these are hybrid methodologies.

A pure algorithm refers to a single algorithm or technique applied to solve a problem from start to finish. In contrast, a hybrid algorithm integrates multiple algorithms or techniques from different domains to solve a problem or improve the performance of a single algorithm. By integrating different methodologies, hybrid methods can leverage each algorithm's strengths and mitigate their limitations (Goldberg, 1989; Wolpert & Macready, 1997). Thus, a hybrid algorithm can be considered a fusion of ideas and methodologies to explore the potentialities of different approaches while compensating for their weaknesses. Thus, by merging complementary algorithms, it is possible to exploit their strengths and overcome limitations, leading to improved overall performance.

A hybrid algorithm combining optimization and machine learning techniques is an effective strategy that uses the advantages of both methodologies to provide a powerful framework for tackling complex problems. This fusion allows for enhanced decision-making capabilities by integrating optimization techniques into the machine learning process and vice-versa. Thereby, a hybrid algorithm can leverage optimization capabilities to guide the learning process and enhance the accuracy and efficiency of decision-making. This integration enables the algorithm to leverage explicit mathematical optimization techniques and data-driven learning capabilities, leading to more effective and efficient decision-making.

This paper describes and explores the main characteristic of numerical optimization and machine learning methods. Through a systematic literature review, an analysis is made of the evolution of non-linear optimization and machine learning methods. In addition to identifying and exploring the main characteristics of each methodology when combined, they can minimize the obstacles and enhance one or both methodologies through hybrid methods. In this way, it is expected to identify the most suitable combinations that result in hybrid methods, considering the algorithm of a machine learning task inspired by optimization strategies or vice versa. Although this work presents the different types of machine learning in detail, the literature review will be restricted to algorithms that perform the classification or clustering task, whether through supervised or unsupervised learning methods, as a large amount of data was identified (papers published), and it was necessary to establish some restrictions to perform a suitable literature review.

In this way, this paper makes a significant contribution by systematically identifying and analyzing the existing knowledge on hybrid algorithms that combine optimization and machine learning. It not only highlights research gaps in the development of hybrid strategies, but also provides insights into future directions. Additionally, the paper presents a comprehensive SWOT analysis of the top ten most cited algorithms in the collected database. This analysis sheds light on the strengths and weaknesses of these algorithms while exploring the opportunities and threats they present. Consequently, the paper offers a thorough understanding of the characteristics of these algorithms, serving as a valuable source of inspiration for future research endeavors. It is crucial to emphasize that while most literature review studies focus on specific algorithms or categories of algorithms, this work aims to encompass a broader scope through the assessment of hybrid methods.

This paper is organized as follows: after the introduction, in Sect. 2, an overview of the numerical optimization method is presented, with the mathematical modeling definition and the main methods and algorithms of global optimization. In Sect. 3, the classes of machine learning techniques are presented, the forms of supervised, unsupervised, and reinforcement learning, as well as the main algorithms belonging to each class. The methodology and the protocol definition to perform the systematic literature review are described in Sect. 4. Section 5 presents the core of this work; it is the systematic literature review, considering a numerical analysis of the papers published since 1970 involving optimization and machine learning; and a bibliometric and in-depth analysis of relevant papers over the three last years is also presented, which encompassed 1007 papers. Furthermore, Sect. 6 presents a SWOT analysis of the ten most cited algorithms of the database collected in the literature review. Finally, Sect. 7 concludes the paper, summarizing the main results achieved and proposing some steps for future works.

2 Numerical optimization

Numerical Optimization is an area of mathematics that studies the identification of extreme points of functions, whether maximum or minimum. In the last decades, the interest in optimization methods has increased mainly due to computational advances and their popularization. Thence, the use of optimization methods is becoming an essential tool for management, decision-making, improvement, and development of technologies to enable efficient systems (Mitchell, 1996).

2.1 Optimization methods

While there are numerous methods and algorithms for handling optimization issues currently, none of them are universal or perfect that can effectively solve all such problems. Generally, each algorithm has particularities, being more appropriate to solve a set of problems according to its characteristics. The problem formulation, the methods, and algorithm choice are critical steps in solving an optimization problem since some methods and algorithms are more appropriate than others depending on the problem.

The first step in an optimization problem is the mathematical model definition. A mathematical model aims to describe a real-world problem into a mathematical function that can be used in optimization algorithms. An optimization problem can be translated into a mathematical language by a set of variables and numerical relations that describe an abstraction of the problem and find the best choice (optimal solution) in R^n from a set of candidate choices. In this way, to develop a mathematical model four steps are necessary to be followed (Rao, 2009; Sohrabi & Azgomi, 2020):

1. Define the decision variables $x_1, x_2, ..., x_n$.
2. Build the objective function $f(x)$, or the set of objectives function $f_1(x), f_2(x), ..., f_k(x)$, that are based on the decision variables and returns a real value.
3. If necessary, define a set of equality and/or inequality constraints, $g_i(x) = 0$ and $h_j(x) \leq 0$ for $i = 1, ..., n_g$ and $j = 1, ...n_h$, that should hold on the decision variables.
4. Characterize the domain sets $D_1, D_2, ..., D_n$ as the domains of the decision variables $x_1, x_2, ..., x_n$, respectively.

The objective function and the constraints on the variables define the optimization problem. The objective function expresses the objective of the problem, which can be any quantity or combination of quantities that may be represented by a single number as people, time, materials, energy, etc; whereas the limitations (constraints), directly affect the decision space and results, providing limitations (restrictions) in the algorithm choices (Gandomi et al., 2013; Nocedal & Wright, 1999; Rao, 2009). Thereby, the main goal of an optimization problem is to minimize or maximize the objective function considering the constraints. Moreover, one possible way to divide the optimization modeling, according to the number of objectives involved, is into two major groups, *Single-objective* and *Multi-objective* (Rao, 2009; Miettinen, 1998) optimization problems. Single optimization problems normally involve a single-objective with or without constraints, whereas multi-objective is an area of multiple-criteria decision-making, involving more than one objective function to be optimized simultaneously, with or without constraints. After the development of the mathematical model, it is necessary to identify the most appropriate method to determine the optimal solution.

A solution to an optimization problem can be described in terms of local and global optimization, and also, the algorithms search can follow similar nomenclature, local search, and global search. Local and global search optimization algorithms are used in different situations or to answer different optimization questions. In local optimization, the goal is to find a solution that minimizes (or maximizes) the objective function in a specific region of the search space, which means that it finds a local solution among feasible points that are in a neighborhood (Boyd, 2004). This kind of search does not guarantee an objective value lower (or higher) than all other feasible points (Boyd, 2004). On the other hand, global optimization tries to find the point that minimizes (or maximizes) the objective function

over all possible feasible solutions. However, the complexity of global optimization methods grows exponentially with the size of the problems (Boyd, 2004).

In optimization, it is possible to use different optimization techniques and algorithms to find the problem solution. Some optimization methods present distinct behavior in the search for the solution to the same problem, so one method can be faster and more reliable in response accuracy than others. There is no single method available for solving all optimization problems efficiently (Rao, 2009), what we have are several algorithms that may be used to solve optimization problems inside their capabilities. In general, the optimization methods can be divided into two large families: *Deterministic* and *Stochastic* approaches that branch into other subfamilies (Heuristics and Metaheuristics), according to particular characteristics, as presented by Fig. 1.

2.2 Deterministic and stochastic approaches

Deterministic methods are based on systematic expressions and theorems that calculate the problem solution whenever it exists. A method is considered deterministic if all of its algorithm steps are well defined under the same initial conditions (Gandomi et al., 2013; Nocedal & Wright, 1999; Rao, 2009). Deterministic methods search the all space of feasible solutions and guarantee the optimality, in a classical point of view (Gandomi et al., 2013; Čorić et al., 2017). Furthermore, in some cases, they demand high computational cost and search time, which makes these methods unable to solve problems in non-polynomial time, NP-hard problems. Deterministic methods show excellent results when the search space is convex and continuous (Giuzio, 2017). When
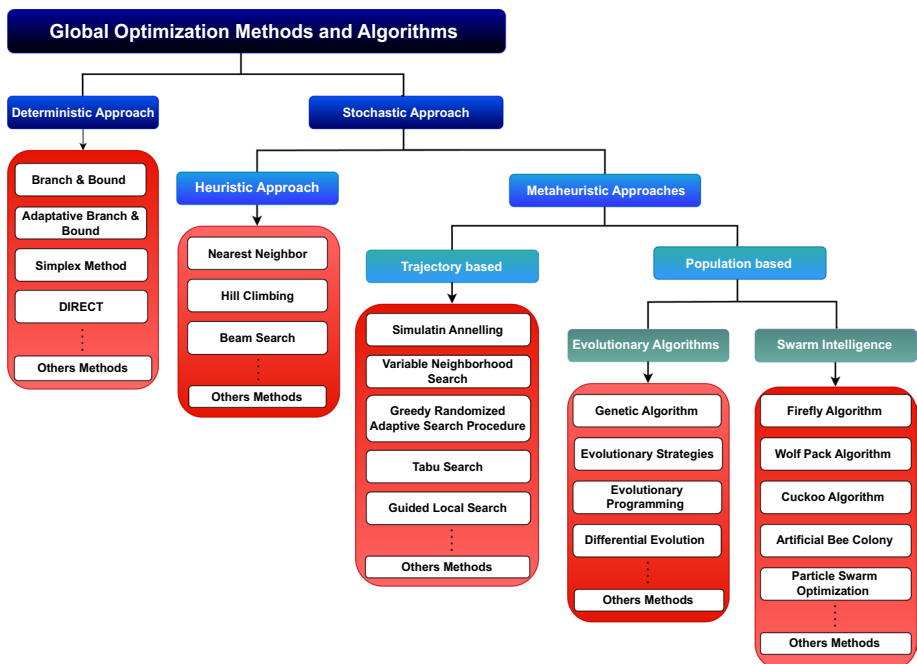

Fig. 1 Optimization methods

these conditions are not guaranteed, these methods do not work as well, provide unacceptable solutions, or do not provide the required degree of accuracy (Zgurovsky et al., 2021). Some examples of deterministic methods can be: Branch and Bound Algorithm (Morrison et al., 2016), Adaptive Branch and Bound Algorithm (ABB), Simplex Algorithm (Simplex) (Nash, 2000), DIRECT algorithm (Jones & Martins, 2021; Jones et al., 1993), among others (Gandomi et al., 2013; Sivanandam & Deepa, 2008; Čorić et al., 2017).

Stochastic methods consist of analyzing and transitioning probabilistic rules and using random values in its procedures (Sivanandam & Deepa, 2008). The stochastic methods are not always able to find the optimal solution with good accuracy, and, in some problems, the optimal solution is slightly different in each algorithm execution. Thus, stochastic optimization methods use uncertainty quantification to produce solutions that optimize the problem (Nocedal & Wright, 1999). Generally, these methods do not guarantee the exact or a high precision solution. However, they usually present a satisfactory solution, very close to the optimal solution, in a shorter time. Stochastic methods are usually divided into two large groups, Heuristic and Metaheuristic, as presented in the next section.

2.3 Heuristic and metaheuristic approaches

Inside the Stochastic family, there are Heuristic and Metaheuristic approaches. According to Yang (2010a) *Heuristic* means "to find" or "to discover by trial and error" (Khattar et al., 2019). On the other hand, Metaheuristic, *meta*-means "beyond" or "higher level", and they generally perform better than simple heuristics (Yang, 2010a). Although the authors (Yang, 2010a) made these definitions, they assume that the terms "heuristic" and "metaheuristic" are sometimes used interchangeably in the literature. The performance of heuristic methods is strongly dependent on the problem type, whereas metaheuristics can be weak depending on the type of problem (Suyyagh et al., 2016). Hill Climbing (HC), Best First Search (BFS), Nearest Neighbor (NN) (Monnot, 2016), Beam Search (Medress et al., 1977), First In First Out (FIFO), and Best Fit (BF) are examples of algorithms based on heuristics (Khattar et al., 2019).

In turn, the metaheuristics can be classified in several ways, one of which is to classify them into *population-based* and *trajectory-based*, as proposed by Yang (2010a). The trajectory-based or point-to-point metaheuristics use a single vector that moves through the design space or search space in a piece-wise style. A better move is always accepted, while a not-so-good move can be accepted with a certain probability. The steps or moves trace a trajectory in the search space, with a non-zero probability that this trajectory can reach the global optimum (Yang, 2010a). Simulating Annealing (SA) (Kirkpatrick et al., 1983), Variable Neighborhood Search (VNS) (Mladenović & Hansen, 1997), Greedy Randomized Adaptive Search Procedure (GRASP) (Deshpande & Triantaphyllou, 1998), Tabu Search (TS) (Glover, 1986) and Guided Local Search (GLS) (Voudouris & Tsang, 1996) are some examples of metaheuristics based on trajectory algorithms.

On the other hand, the population-based metaheuristics use multiple candidates as search points, and the population characteristics are used to guide the search (Khattar et al., 2019). Population-based metaheuristics can be divided into Evolutionary and Swarm Intelligence algorithms in a classical classification (Khattar et al., 2019).

2.3.1 Evolutionary computation

The term "Evolutionary Computation" (EC) refers to a variety of problem-solving methods based on biological evolution concepts that can simulate the process of natural selection throughout the search process (Bansal et al., 2019). The strategy used by evolutionary algorithms is to try to find global optimal solutions through genetic operators, such as selection, crossover and mutation (the most common), applied to a set of dynamic individuals, denoted population (Sivanandam & Deepa, 2008). The evolutionary computation algorithms start by producing a set of random candidate solutions (population). The population is represented by individuals arranged in the search space, which is the space where each variable can have values (some examples are $\mathbb{Z}^n$, $\mathbb{R}^n$, $\{0, 1\}^n$, ...). The search space is delimited by the domain of the objective function (and constraints, if any), which ensures that all individuals are admitted as solutions to the problem (Sivanandam & Deepa, 2008). By iteratively applying genetic operators, the population changed to produce new feasible solutions. This process stochastically discards poor solutions and evolves into more suitable (better) solutions (Bansal et al., 2019). Due to the very nature of these operators, which are based on Darwin's evolution principles (in which the most adapted individuals of a given population survive, whereas the less adapted die to be replaced by their offspring (Bansal et al., 2019; Sivanandam & Deepa, 2008)), evolved solutions are expected to become better generation by generation (iteration). Like any iterative process, the evolutionary algorithms require a stopping criterion to interrupt the search and define the optimal solution (Sivanandam & Deepa, 2008). Some examples of stopping criteria are described in Azevedo (2020).

In the class of evolutionary algorithms, four families deserve to be highlighted: Genetic Algorithms (GA) (Holland, 1992), which is based on the Darwinian principle of survival of the fittest and encoding of individuals; Evolutionary Programming (EP) (Jacob, 2001) and Differential Evolution (DE) (Storn & Price, 1997), which are inspired by the theory of evolution using natural selection; Evolutionary Strategy (ES) (Jacob, 2001), which is a search technique based on the idea of adaptation and evolution and Differential Evolution.

2.3.2 Swarm algorithms

Swarm algorithms are derived from observations of nature's methods for optimization. Animals, for instance, have a natural ability to develop ways to use less energy to carry out survival-related functions like protection, defense, migration, localization, and food digestion. Thus, swarm intelligence studies the collective behavior that emerged from social insects or animals working under very few rules (Bansal et al., 2019). Furthermore, the term swarm algorithm is a class of algorithms that study and implement the behavior of social entities in artificial models and systems. Technically, the term swarm refers to a collection of interacting homogeneous agents or individuals; in other words, it can be defined as a collection of individuals or objects in disorganized movements, such as insects, birds, and fishes (Bansal et al., 2019).

The swarm algorithm is characterized by two phases, variation and selection, responsible for maintaining the balance between exploration and exploitation and forcing the entire swarm, i.e., the set of potential solutions, to update their positions. In the variation phase, the different areas of the search space are explored, and the selection phase is responsible for exploiting the previous experiences (Bansal et al., 2019).

In swarm intelligence concepts, individual population members have an identity, which they retain over time in the form of temporarily linked movements (Bansal et al., 2019). A group of homogeneous agents exhibits swarm intelligence if and only if they present two characteristics: *self-organization* and *division of labor* (Karaboga, 2005). Self-organization is responsible for providing diversity and exploitation in the swarm intelligence, a fluctuation that provides new situations in the process and helps to eliminate stagnation. Furthermore, it is responsible for multiple interactions, which is the way to learn from more than one individual within society and improves the overall intelligence of the swarm. The division of labor helps different tasks to be performed simultaneously and makes the swarm capable of handling changed conditions in the search space (Bonabeau et al., 1999).

Currently, there is a long list of popular and successful swarm intelligence algorithms, the most popular being: Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995) which is inspired by the behavior of birds flocking or fish schooling; the Ant Colony Optimization (ACO) (Dorigo & Stützle, 2003) which is inspired by foraging behavior of ants; Artificial Bee Colony (ABC) (Karaboga, 2005) which investigates the behavior of honey bees; Wolf Pack Search (Yang et al., 2007) which simulates predator behavior and prey distribution of wolves; Firefly Algorithm (Yang, 2010b), inspired by the flashing behavior of fireflies, among many others.

3 Machine learning

Learning through experience and personal knowledge, which is spread from generation to generation, is at the heart of human intelligence (Theodoridis, 2015). Machine learning addresses how to build computer programs (learning systems) that improve their performance with experience (Gopal, 2018). In this case, the experience is provided by a data analysis process performed by a specialized algorithm. Thus, the machine learning method applies algorithms to extract patterns from a set of data by using mathematical, statistical, optimization, and knowledge discovery methods (Bishop, 2006)

Although the machine learning term was coined around the 1960s (Mitchell, 1996; Rao, 2009), it only gained popularity in the 21st century due to the advancement of computational resources. Pattern recognition (classification), numerical prediction (regression), clustering, optimization, and control are typical issues that machine learning frequently addresses (Gopal, 2018). Nowadays it is possible to find applications in practically every area of science, as example: music (Bressan & Azevedo, 2017), health (Abarghouei et al., 2009), economics (Cicceri et al., 2020; Wuerges & Borba, 2010), industrial segments (Bressan et al., 2021; Fávero & Zoucas, 2016), education (Agrusti et al., 2019; Azevedo et al., 2022; Zhu, 2019), among many others.

Three forms of machine learning are considered in Bishop (2006): Supervised Learning, Unsupervised Learning, and Reinforcement Learning. Within each class, it is possible to distinguish methods according to the way they obtain the knowledge, be it classification, regression, clustering, learning of associations, relations, differential equations, among others (Kononenko & Kukar, 2007). Figure 2 presents the machine learning methods, as considered in Bishop (2006).

Machine learning aims to build a hypothesis (model) with the ability to extract the information presented in the training data and generalize the acquired knowledge to unknown samples. This model must be simple in terms of complexity and good in terms of empirical errors in the data (Gopal, 2018). Before using a machine learning system,
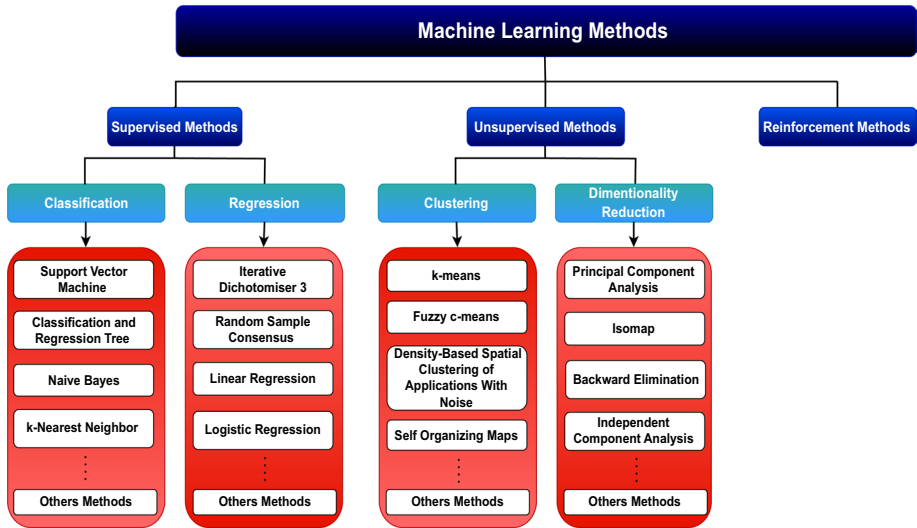
Fig. 2 Machine learning methods

it is necessary to assess the performance of the model schema (error rate) on new data. The techniques chosen for this assessment are related to the size of the available data set. When the data set is large enough, it is common to consider three independent data sets: the training set (which is used to build the initial model), the validation set (which is responsible for optimizing the model, adjusting the initial to be more general), and the test set (which computes the error rate of the final model) (Gopal, 2018). It is important to highlight that these three sets must be selected independently, so a large enough data set is required.

However, in some cases, mainly in practical situations, it is necessary to deal with limited data, as it is not possible to obtain three significant and independent data sets. Thus, other evaluation techniques should be used. One possibility is to use the holdout technique. In this case, a certain amount of data is intended for testing, while the rest is employed for training. It is common to hold out a third of the data for testing and use the remaining for training.

Another possibility, for small data sets, is k-fold cross-validation. This technique is very useful in fixed data samples to forecast the success rate of a learning method. In k-fold cross-validation, the training and testing process is done k times. Thus, consider a given data D, which is randomly divided into k mutually exclusive subsets D_k, in which $k = 1, ..., k$ each of approximately equal size. In the iteration k, the D_k partition is reserved for testing, and the remaining subsets are used to train the model. Thus, in the first iteration, the set $D_2 \cup D_3 \cup ... \cup D_k$ serves as the training set to attain the first model, which is tested on D_1; the second iteration is trained on $D_1 \cup D_3 \cup ... \cup D_k$ and tested on D_2; and so on. In the end, the k error estimates received from k iterations are averaged to give rise to an overall error estimate. So, k equals to 10 is the standard number used to predict the error rate of a learning method (Gopal, 2018).

In the following, the three forms of machine learning previously referred are described: Supervised Learning, Unsupervised Learning, and Reinforcement Learning.

3.1 Supervised learning methods

Supervised learning methods attempt to discover the relationship between input attributes (independent variables) and a target attribute (dependent variable) (Rokach et al., 2005). Mathematically, in supervised learning, the method is designed to explore the known *a priori* information.

Consider the data set *DS* used to infer a model of the system, in which each individual instance is represented by x^i (Gopal, 2018), given by

- N is the number of data set elements,
- n_f is the number of attributes (features) of each instance x^i,

$$DS = \begin{bmatrix} x^1, y^1 \\ ... \\ x^N, y^N \end{bmatrix} \tag{1}$$

The data set *DS* lies in state space $\mathbb{R}^N \times \mathbb{R}^{n_f+1}$. The choice of features (or attributes, or parameters) $x^i_j; j = 1, ..., n_f$, for a given instance i, significantly affects the output (Gopal, 2018). There are two types of tasks for which supervised learning is used: pattern classification or regression (whose purpose is to predict the value of one or more target attributes).

3.1.1 Classification

Consider the output vector $y \in Y$ where Y represents M discrete classes. The task of the classifier is to categorize the data into different classes (i.e., to decide which of the M classes each new vector x^{new} belongs to) based on the training of the experiments presented in the training set X (Gopal, 2018). Several algorithms can be applied in the classification task, such as Decision Tree (Rokach & Maimon, 2008), Support Vector Machine, k-Nearest Neighbor, Naive Bayes (Neapolitan, 2003), among others (Cichosz, 2015; Han et al., 2012; Theodoridis, 2015). Figure 3 illustrates a general classification algorithm procedure whose objective is to classify a set of points (described by the input vector) into different classes (blue and orange) as defined in the input classes vector.
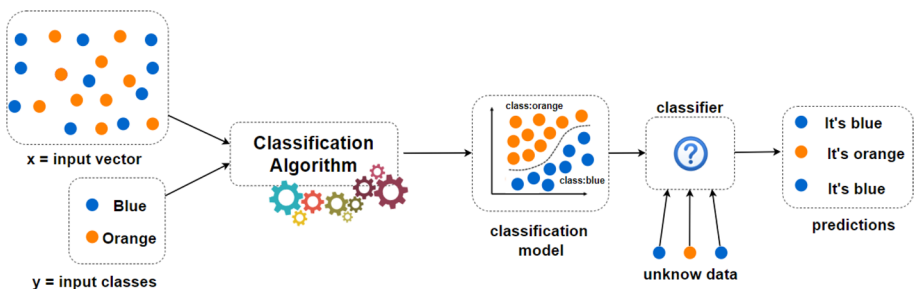

Fig. 3 Classification algorithm procedures

3.1.2 Regression

Regression learning aims to explore the relationship between independent variables or features (input variables *x*) and a dependent variable or outcome (continuous output variable *y*). The regression task consists of fitting a function to the input-output data in order to predict (numeric) output values for new inputs (Gopal, 2018). There are several forms of regression, such as linear, multiple, weighted, polynomial, nonparametric, and robust (Han et al., 2012). Simple Linear Regression, Logistic regression, Multivariate Regression, and Regression tree are some examples of algorithms that can be used to build regression models (Bishop, 2006; Gopal, 2018; Han et al., 2012). Figure 4 illustrates a general linear regression algorithm procedure, in which the aim is to define a linear function that represents the data set behavior.

3.2 Unsupervised learning methods

Sometimes, there is no information about the relationship between input and output attributes in machine learning problems. Thus, the algorithms must discover similarities or dissimilarities in the data set. The method requires more human understanding than supervised techniques, since a decision-maker, that is, a person or a group of people, are responsible for the final decision-making.

Although supervised and unsupervised methods work with data and require exploration and understanding of the data regarding the application domain, there are some crucial differences involving both methodologies. The main difference is the absence of an output vector of the target variable, as in supervised methods. In addition, unsupervised learning is often associated with creative endeavors-exploration, understanding, and refinement, which do not lend themselves to specific procedures, as supervised methods. For this reason, it cannot be automated. Moreover, there is no right or wrong answer and no simple statistical measure that summarizes if the findings are excellent or bad. Instead, descriptive statistics and visualization are key parts of the process (Gopal, 2018). Thereby, unsupervised learning is thus typically split into Clustering methods and Dimensionality Reduction methods.

3.2.1 Clustering methods

As previously mentioned, in some cases, the data set is not labeled, so it is necessary to analyze intrinsic characteristics of each element of the data set. Among the unsupervised methods, clustering techniques can be considered the most popular (Kononenko & Kukar, 2007). Basically, clustering can be defined as a task of grouping a set of elements with
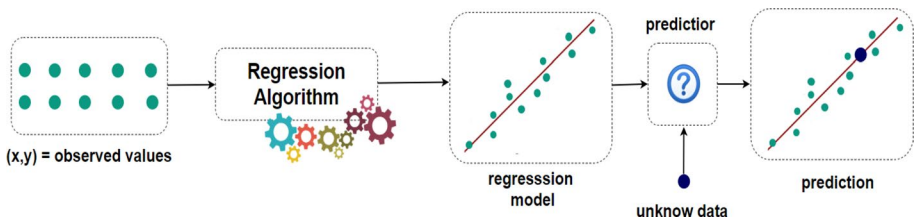

Fig. 4 Regression algorithm procedures

similarities in the same group and with dissimilarities in other groups (Iglesias et al., 2021; Shalev-Shwartz & Ben-David, 2014). This procedure is very useful in engineering, health science, humanities, economics, and other areas (Albarakati & Obradovic, 2019; Azevedo et al., 2022; Bi et al., 2020; Chen et al., 2020; Zhou et al., 2019).

The evaluation of a data set's constituent members' proximity and the division of the data set into groups while taking into account the similarity and dissimilarity between a pair of elements are both essential steps in the clustering process. It is useful to denote the distance between two instances x^i and x^j as $d(x^i, x^j)$ to quantify the similarity between them.

To define the quality of the cluster, it is necessary to use an evaluation criteria measure that is usually divided into two categories: internal and external. The internal quality metrics usually measure the compactness of the clusters using some similarity measure. And, the external measures can be useful for examining whether the structure of the clusters matches some predefined classification of the instances (Rokach et al., 2005). According to Estivill-Castro and Yang (2000) the notion of "cluster" is not precisely defined, for this reason, many clustering methods and algorithms have been developed. These methods can be divided into 5 categories (Mehta et al., 2020): Partitioning based, Hierarchical based, Density-based, Grid-based and Model-based. Following are some examples of clustering algorithms belonging to different clustering method categories (Kononenko & Kukar, 2007; Mehta et al., 2020; Rokach et al., 2005): k-means Algorithm, Fuzzy c-means Algorithm (FCM), Clustering Using Representatives (CURE), Density-Based Spatial Clustering of Applications with Noise (DBSCAN), Ordering Points to Identify the Clustering Structure (OPTICS), Optimal Grid-Clustering (OptiGrid), Gaussian Mixture Model Clustering, Self-Organizing Maps Clustering (SOMs).

Figure 5 illustrates an exclusive clustering algorithm procedure, whose objective is to divide the data set into groups according to the data characteristics. In this case, the term exclusive is associated with the idea that each data point exclusively belongs to one cluster.

3.2.2 Dimensionality reduction

In machine learning, the number of input variables of a data set is referred to as its dimensionality. Thus, reduction dimension techniques refer to selecting the most relevant variables and leaving out the irrelevant ones that can confuse, deteriorate and slow down the mining process. However, the selected variables must keep as much variation as possible in the original data set (Gopal, 2018). The reduction dimension methods can be categorized based on two factors (Gopal, 2018):
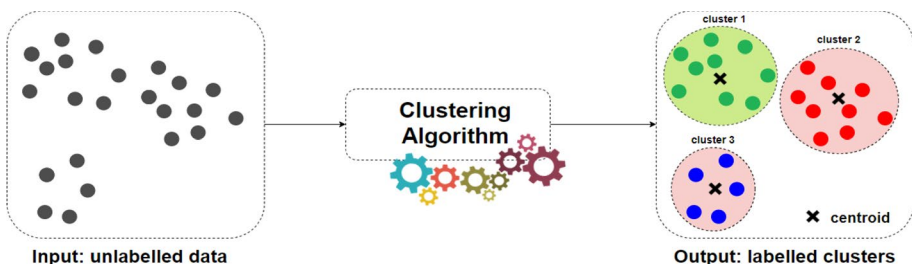

Fig. 5 Exclusive clustering algorithm procedures

- The first factor is whether the technique employs the target variable to select input variables or not.
- The second factor is whether the technique utilizes a subset of the original variables or derives new variables from them to maximize the amount of information.

According to Gopal (2018), the benefit of maintaining the original variables is understandable because the original variables in the data are easier to understand than those generated automatically by some variable reduction technique. On the other hand, in some cases involving big data set, applying reduction dimension methods is the only way to guarantee an efficient machine learning process. Principal Components Analysis (PCA) (Gopal, 2018), Rough Sets-Based Attribute Reduction (Gopal, 2018), Isomap, Backward Elimination, and Independent Component Analysis (ICA) are examples of dimensionality reduction techniques (Velliangiri et al., 2019).

3.3 Reinforcement learning

Reinforcement learning is a machine learning training method based on the rewarding procedure being the most used on dynamic control systems, but it can also be used to solve optimization problems (Kononenko & Kukar, 2007). Reinforcement learning is grounded on the principle that if an action is followed by a satisfactory state of affairs or by an improvement in the state of affairs, then the inclination to produce that action becomes stronger, that is, reinforced (Gopal, 2018).

Reinforcement learning deals with the problem of teaching an autonomous agent that acts and senses its environment to choose optimal actions for achieving its goals (Kononenko & Kukar, 2007). The agent receives information about the current state in the environment, and needs to exploit the knowledge it already possesses from being greedy to maximize reward, but also needs to explore to choose better actions in the future (Gopal, 2018; Kononenko & Kukar, 2007).

With more formalism, reinforcement learning can be formulated as a Markov decision process as presented in Fig. 6. At each time step t, given the current state s_t (and current reward r_t), the agent needs to learn a strategy (i.e. the "value function") that selects the optimal decision or action a_t. The action will have an impact on the environment that induces the next reward signal r_{t+1} (which can be positive, negative, or zero) and also produces the next state s_{t+1}. The reinforcement learning continues with a trial-and-error process until it learns an optimal or suboptimal strategy (Galatzer-Levy et al., 2018).

It is important to clarify that reinforcement learning is not the same as supervised learning. While in supervised learning the data set works as a teacher, teaching the patterns to the algorithms, in reinforcement learning the algorithm learns from past actions, according

Fig. 6 Reinforcement learning schematic. Adapted from Galatzer-Levy et al. (2018)

to critics. The critic gives no advance information. Thereby, after several actions are taken and rewards received, it is desirable to evaluate the individual actions taken and identify the ones that led to the reward after several actions are performed and rewards are received. This allows for the recording and future recall of these movements (Gopal, 2018).

4 Methodology

To perform a systematic literature review, a protocol must be established to define the criteria before the review is conducted. A good literature review must be a comprehensive, transparent search conducted over multiple databases, and its steps can be replicated and reproduced by other researchers. It involves planning a well thought out search strategy that has a specific focus or answers a defined question (Dewey & Drahota, 2016). The following section describes the criterion and strategies used in this paper.

4.1 Objective and research questions

The first step in preparing a systematic literature review is to define the main objectives and the research questions that will guide the research. With this, the focus of the work is traced, as well as the points to be discovered, understood, or studied are defined in their answers. Thus, the main objective of this research is to identify and analyze the combination of machine learning and optimization in hybrid algorithms oriented toward algorithm improvement. Thereby, the defined research questions are:

- **RQ1**: What are the difficulties of classical bio-inspired optimization algorithms and machine learning algorithms?
- **RQ2**: What are the methods or techniques already developed that combine optimization and machine learning in order to improve algorithms performance?
- **RQ3**: What are the potentialities of the optimization and learning algorithm developed?

4.2 Keywords and logic search

The keywords define the main topics that will be dealt with in the papers that will compose the database and will be analyzed. Thus, an initial set of possible keywords is defined and, after an initial search and analysis, the final set of keywords is defined. After that, the logic search is established, as presented below.

Possible keywords: Stochastic Global Optimization; Evolutionary Algorithms; Population Based Methods; Operational Research; Optimization; Heuristic; Meta-heuristic; Convergence; Algorithms Comparisons; Hybrid Algorithms; Machine learning; Data Analyze; Supervised learning/Methods; Unsupervised/Non-supervised Learning/Methods; pattern recognition; classification; clustering.

Keywords selected: Machine learning; Optimization; Swarm; Evolutionary; Classification; Clustering.

Logic search: (Machine learning) AND (Optimization) AND (Swarm OR Evolutionary) AND (Classification OR Clustering) AND NOT (Reinforcement OR Neural Networks OR Ensemble methods OR Regression OR Game theory OR Robotics OR Deep learning OR Dimensionality Reduction).

It is noteworthy that the logical connective AND NOT was used to exclude strongly recurring subjects that were not of interest in this work.

4.3 Source selection

The sources must be available via the web, preferably in scientific databases in the area. Therefore, the source list selected was:

- IEEE Digital Library (http://ieeexplore.ieee.org)
- Scopus Digital Library (http://www.scopus.com)
- WoS Web of Science (https://www.webofscience.com)

The logic search includes the title, abstract, or keywords for the Scopus database and all metadata of the WoS and the IEEE, due to source library restrictions.

4.4 Type of articles

Studies conducted by researchers and developers in the area of optimization and/or machine learning algorithms, as well as practical and relevant applications of hybrid algorithms involving both methodologies. Regarding language, only works published in English were considered.

4.5 Inclusion and exclusion criterion applied for works selection

Inclusion Criterion

- (a) Works published and fully available in scientific databases will be included.
- (b) Recent works (published from 2019 to 2021) will be included.
- (c) Works that address the development of hybrid algorithms will be included, since they involve optimization and machine learning methods to improve one or both methodologies.
- (d) Works that provide practical or theoretical applications of hybrid algorithms will be included, even if they do not focus mainly on the development of these algorithms, but only on methodology application.

Exclusion Criterion

- (a) Works published in short articles, abstracts, or posters will be excluded.
- (b) Works that only present applications of techniques without contributions to the improvement or development of combined methods will be excluded.
- (c) Works that do not use comparison between classic or hybrid methodologies through statistical metrics will be excluded.

4.6 Primary studies selection process

Using the logic search previously described in search engines, the documents data base was generated. After reading the title and the abstract and applying the inclusion and exclusion

criteria, the papers will be selected if their relevance is confirmed by the main reviewer. If there is any doubt about the relevance, the other reviewers will be consulted.

4.7 Strategies for information extraction

After defining the works definitively included, the documents will be read in full. The reviewer will summarize each one, highlighting the methods used to improve methodologies, parameters considered for comparisons, results achieved, and performance evaluation.

4.8 Summary of results

After reading and summarizing the selected works, a technical report will be prepared with a quantitative analysis of the works. A qualitative analysis will also be carried out to define each method's advantages and disadvantages.

5 Systematic literature review

Optimization and machine learning are very broad concepts and have numerous applications, separate or combined. As it is known, with the advancement of computational resources, optimization, and machine learning have gained more popularity, resulting in a large volume of published documents. Figure 7 shows the number of documents published each year, in three databases, Scopus, Web of Science (WoS), and the Institute of Electrical and Electronics Engineers (IEEE), with mention of the word "Optimization" and the word "Machine Learning" in the title, abstract or keywords for Scopus database and all metadata of WoS and IEEE.
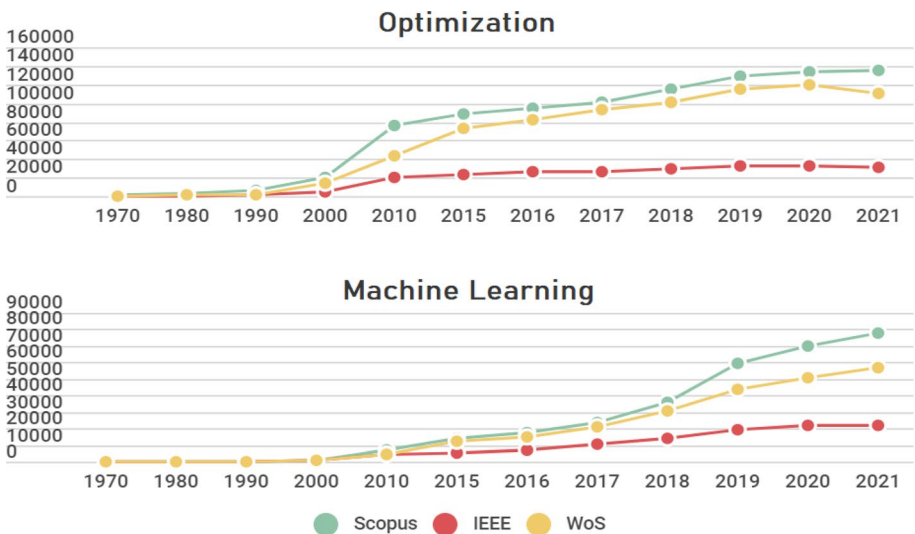

Fig. 7 Number of publications that mentioned the terms "Optimization" and "Machine Learning"

It is essential to highlight that the Scopus and WoS document bases include several publishers. In contrast, the IEEE only completes documents from the IEEE publisher, so the volume of publications is much smaller. Although it would be expected that the most representative documents of the IEEE would be present either in Scopus or WoS, several relevant and not duplicated documents were found. For this reason, it chose to maintain the comparison with the documents of the IEEE base.

From Fig. 7, it is possible to note the impact of computational advances in the area after the 2000 years, as well as the growing expansion to the present day. Moreover, it is important to highlight that optimization is a concept that started gaining attention around the 1990s (although the origin of the term optimization can be traced to the early period of World War II (Rao, 2009)), whereas machine learning became popular around the 2010s. In the same period (the 2010s), bio-inspired optimization algorithms (Swarm and Evolutionary algorithms) become a more popular topic among the scientific optimization community. And, in the last 5 years, there has been a considerable expansion in the studies involving machine learning and bio-inspired methods together, as it is possible to note in Fig. 8.

To carry out a systematic review involving the integration of global methodologies is not an easy task. By the first exploratory search, it was noticed that the available systematic reviews specify in a hard restrict way the main topic inside each methodology, for example: Clustering Evolutionary (Ezugwu et al., 2020, 2021), Evolutionary Decision Tree (Barros et al., 2012), Artificial Bee Colony (Karaboga et al., 2020), Multi-Objective Optimization Problems With Irregular Pareto Fronts (Hua et al., 2021). Thereby, due to the large amount of documents, to carry out this literature review it was necessary to establish a strict technical protocol to filter the most relevant information, as described in the next section.

Thus, after a preliminary study, through an explanatory analysis, the following logic search was defined: "(Machine learning) AND (Optimization) AND (Swarm OR Evolutionary) AND (Classification OR Clustering) AND NOT (Reinforcement OR Neural
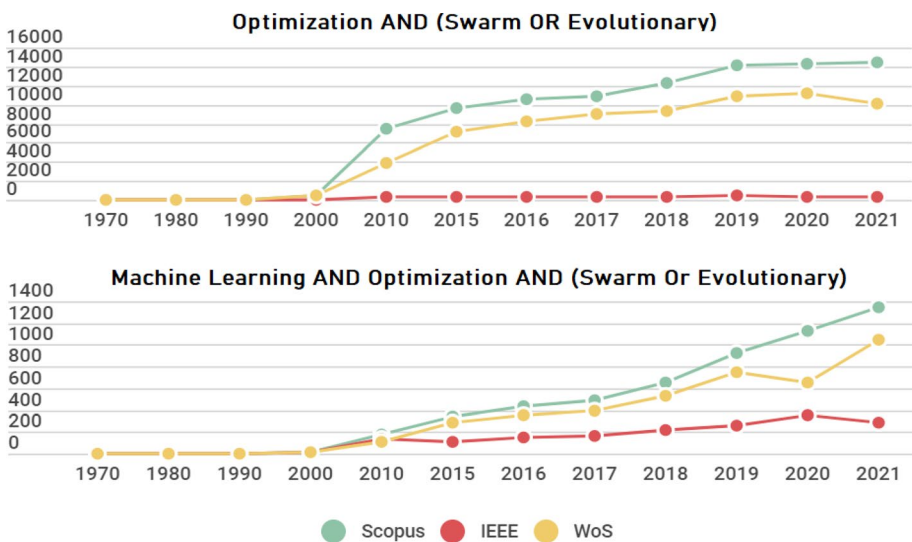

Fig. 8 Number of publications that mentioned the terms "Optimization AND (Swarm or evolutionary)" and "Machine Learning AND Optimization AND (Swarm or evolutionary)"

Networks OR Ensemble methods OR Regression OR Game theory OR Robotics OR Deep learning)". This logic resulted in 4362 documents (2569 from Scopus, 662 from IEEE, and 1131 from WoS). Due to the large number of documents, the search period was limited from the years 2019 to 2021. Only the documents written in English were considered, and the posters and short or abstract papers were filtered and removed from the database. These restrictions were equally used on the sources Scopus, IEEE, and WoS sources. Thus, 1129 documents remained in the database (479 from Scopus, 200 from IEEE, and 450 from WoS). 134 duplicate works were identified, leaving 995 works to be analyzed by the inclusion and exclusion criteria previously presented. Besides, 12 other relevant documents were added to the final database, totaling 1007 works. This systematic categorization of the documents is illustrated in Fig. 9.

5.1 An overview of database documents

In order to analyze the dynamics and evolution of scientific information regarding the combination of the terms optimization and machine learning through the logic search defined above, the 1007 articles from the database were analyzed using the Bibliometrix tool in Rstudio software (Aria & Cuccurullo, 2017).

Firstly, the word cloud limited to the 50 most cited keywords provided by the authors was analyzed, as shown in Fig. 10. Note that the words Machine Learning, Classification, and Optimization are highlighted and, among the algorithms, the Particle Swarm Optimization, Support Vector Machine, and the Genetic Algorithm stand out.

Figure 11 shows the countries of the corresponding authors, where the red bar represents Multiple Country Publications (MCP) and the blue bar Single Country
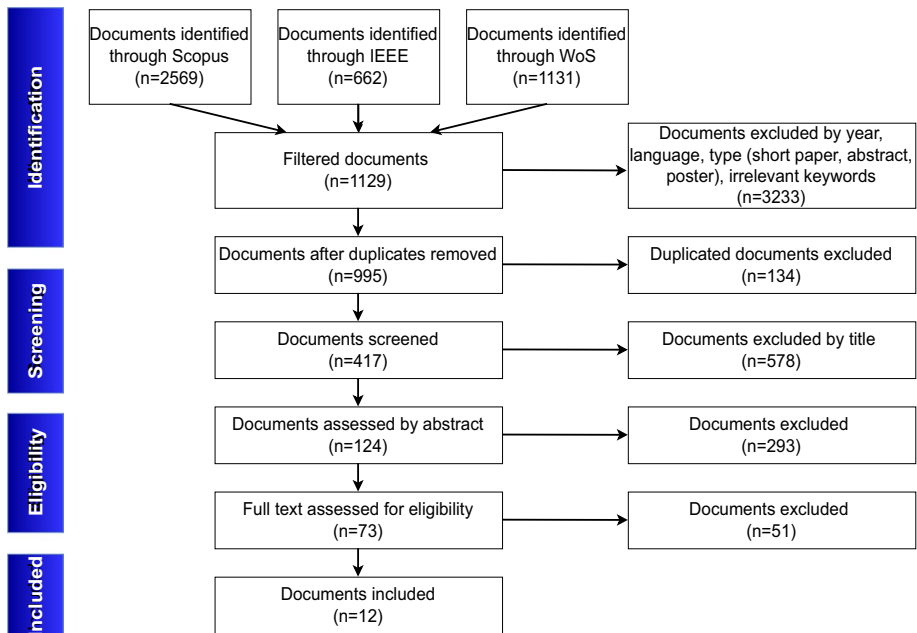

Fig. 9 PRISMA diagram of the literature review process (Page et al., 2021)

Fig. 10 Word cloud limited to the 50 most cited keywords

Fig. 11 Corresponding author's country

Publications (SCP). As was to be expected, the predominance of Chinese authors is noted as the ones who most contribute to the subject. Moreover, there is a strong presence among Indian and US authors.

Figure 12 illustrates the author's co-citation network, which allows the identification of relationships between authors by determining which authors cited other pairs of authors. In this case, it is possible to highlight three names as the most cited in each
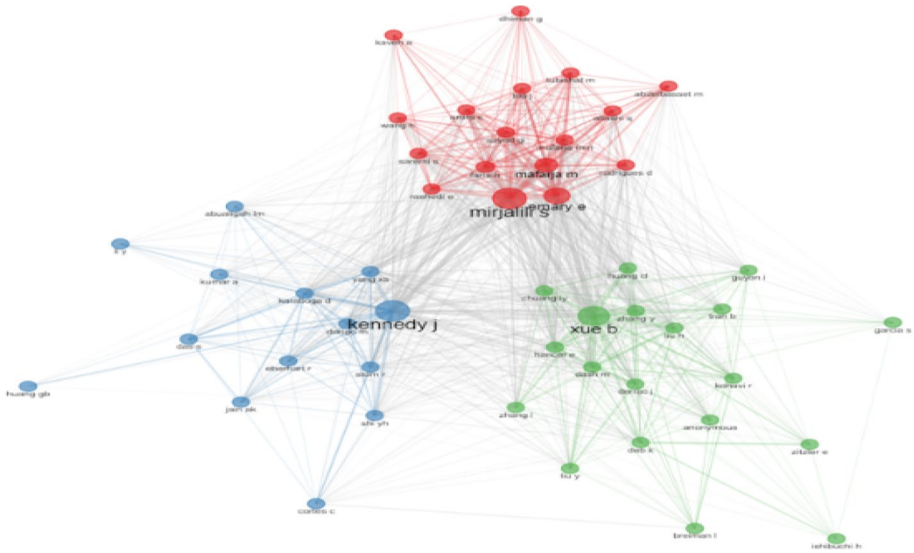
Fig. 12 Authors co-citation network

cluster: Mirjalili, S. as the most co-cited in the red cluster, Kennedy, J., in the blue cluster, and Xue, B. the most co-cited author in the green cluster.

5.2 State-of-art

After applying the inclusion and exclusion criterion, in the title and abstract, 73 works were selected to be read and analyzed in more depth.

Even with several optimization algorithms, new algorithms are continuously being developed. This process is due to the No-Free-Lunch theorem, established by Wolpert and Macready in 1997 (Wolpert & Macready, 1997). According to them, if an algorithm *A* can outperform another algorithm *B* on some set of optimization problems, then there are some other functions in which *B* will surpass *A*. If their performance are averaged on all possible functions, *A* and *B* can perform equally well on all these functions (Wolpert & Macready, 1997; Yang et al., 2020). In other words, the main idea of this theorem is that some algorithms can be better than others on the same problem. Thence, the benchmark studies are dedicated to establishing performance measures to assist the decision-making of the most suitable optimization algorithm for the problem and to establish a mechanism to test and validate new algorithms.

The optimization procedure can be included in a part of machine learning algorithms (Gopal, 2018). Thus, the evolution of optimization algorithms also contributes to the evolution of the performance of machine learning algorithms. Every algorithm has its strengths and weaknesses, whether it is focused on optimization or machine learning. Thus, as there is no perfect algorithm, and mainly to deal with the weaknesses, researchers began to develop hybrid methodologies that combine ideas from different search paradigms and/ or completely different algorithms. In this case, the main idea is to achieve some synergistic behavior whose combination with other techniques compensates for the deficiencies

of a particular technique, and its advantages are enhanced due to this same combination (Cotta et al., 2018).

In this section and also in the next one, the most relevant papers found in the selected database are better analyzed. The papers were divided into two sections, "Clustering Methods" and "Supervised Classification Methods" since this work aims to analyze these methods deeply. Table 1 presents a systematic categorization of the papers identified in the literature review performed for clustering and classification tasks.

5.2.1 Clustering methods

Clustering is one of the most used methods for unsupervised learning. The most well-known clustering methods are based on distance measures, distance metrics, and similarity functions. Its main disadvantage is getting stuck in the local optimum; moreover, its performance strongly depends on the initial values of the cluster centers (Eesa & Orman, 2020). The k-means (a partitioning clustering algorithm), is one of the most popular clustering algorithms and is an example of an algorithm dependent on the initial solution. In consequence, several studies propose using nature-inspired metaheuristics to find a solution that maximizes the separation between different clusters and maximizes the cohesion between data points in the same cluster (Qaddoura et al., 2021). Zhou et al. (2019) describe the use of the Symbiotic Organism Search (SOS) to solve the problem, while Eesa and Orman (2020) present a bio-inspired Cuttlefish Algorithm (CFA) to search for the best cluster centers that can minimize the clustering metrics. An approach based on a Whale Optimization Algorithm (WOA) is suggested by Singh (2021). In turn, Singh and Kumar (2020) uses a modified Cat Swarm Optimization (CSO) to improve the clustering algorithm performance. Qaddoura et al. (2021) developed an evolutionary algorithm based on the evolutionary behavior of a genetic algorithm combined with the nearest neighbor search technique for clustering problems. Nemmich et al. (2019) used the Bees algorithm with a Memory Scheme to solve data clustering problems. All approaches were tested on several benchmark data sets and sometimes real-life problems, and the authors considered various statistical tests to justify the effectiveness of the suggested approaches.

Pacifico and Ludermir (2021) proposed a hybridization between Self-adaptive Particle Swarm Optimization (IDPSO) and the k-means algorithm, in which the IDPSO is used in the exploration phase, and the k-means is adopted in the exploitation phase of the algorithm. The self-adaptive scheme is employed so that the parameters for each individual of PSO may reflect the current state of the search promoted by the entire population. The approach also uses a crossover operator to improve the diversification of the PSO population, avoiding premature convergence.

El-Shorbagy et al. (2019) proposed an enhanced Genetic Algorithm with a new mutation operator based on the k-means algorithm for cluster analysis. In this case, the population of GA is initialized by the k-means algorithm to reach the best cluster centers; thereafter, the GA operators are applied with a new mutation strategy that depends on the extreme points in the cluster groups.

Atabay et al. (2016) introduced a clustering algorithm that combines the PSO and k-means algorithms. This integration resolves the sensitivity of k-means to the initial choice of centroids. Additionally, the algorithm utilizes the rapid convergence ability of k-means by transitioning the cluster center from the previous location to the average location of the points belonging to that cluster in each iteration. This results in accelerated convergence and improved outcomes for the PSO algorithm.

Table 1 Hybrid approach algorithms involving optimization and machine learning

Category	Main problems explored	Hybrid approach	
Clustering	Optimum number of centroids. Exploration and exploitation capacities. Centroids position	k-means &	Ant lion optimization (ALO) (Chen et al., 2020) genetic algorithm (GA) (El-Shorbagy et al., 2019) Particle swarm optimization (PSO) (Pacifico & Ludermir, 2021; Atabay et al., 2016; Ratanavilisagul, 2020) social spider optimization (SSO) (Thalamala et al., 2020) symbiotic organism search (SOS) (Zhou et al., 2019)
Clustering	Fuzzy parametrization. Optimum number of centroids	Fuzzy c-means Clustering &	Ant colony optimization (ACO) (Niknam & Thulasiraman, 2020) Animal behavior global optimization (Absara et al., 2020) artificial bee colony (ABC) (Alrosan, 2021) grey wolf optimizer (GWO) (Edwin Dhas & Sankara Gomathi, 2020) PSO (Nguyen & Kuo, 2019) sine cosine algorithm (SCA), GA and PSO (Kuo et al., 2020)
Clustering	Optimum number of centroids. Exploration and exploitation capacities. Centroids position	Others clustering Algorithms &	Bees algorithm with memory scheme (Nemmich et al., 2019) Cat swarm optimization (CSO) (Singh & Kumar, 2020) cuttlefish algorithm (CFA) (Eesa & Orman, 2020) Firefly algorithm (Ezugwu et al., 2020) flower pollination algorithm (Dhal et al., 2020) GA (Dutta et al., 2019; Qaddoura et al., 2021) GWO and PSO (Zhang et al., 2021) whale optimization algorithm (Singh, 2021)
Classification	Error rate minimization. Accuracy improvement. Overfitting problems.	Decision tree &	ACO, PSO, and BA (Bida & Aouat, 2021) Differential evolution (Jariyavajee et al., 2019) GA (Adibi, 2019; Damanik et al., 2019) PSO (Nagra et al., 2020; Agustina et al., 2019) Salp swarm optimization (Sarala et al., 2021)
Classification	Error rate minimization. Accuracy improvement. Parameters definition	k-Nearest Neighbors &	Chicken Swarm optimization (CSOA), GWO and ABC (Jain et al., 2022) GA (Shih & Ting, 2019) Genetic programming (Huang et al., 2019) PSO (Lee et al., 2020)
Classification	Low performance on unbalanced data set. Convergence velocity	Random Forest &	Firefly algorithm (Farrell et al., 2020; Aslam et al., 2021) GA (Mittal et al., 2020) Kangaroo Mob optimization (Moldovan et al., 2019) PSO (Asadi et al., 2021; Zhou et al., 2020)

Table 1 (continued)

Category	Main problems explored	Hybrid approach	
Classification	Error rate minimization. Accuracy improvement. Parameters definition.	Support vector Machine &	Bat algorithm (Tharwat & Hassanien, 2019) CSOA (Moldovan, 2020) Differential evolution (Dixit et al., 2021) Fireworks algorithm (Dong et al., 2020) GA (Wu, 2019) Harris Hawks optimization (Houssein et al., 2021) Horse optimization algorithm (Moldovan et al., 2020) PSO (Moodi et al., 2021; Zhao et al., 2019; Yu et al., 2021; He & Fu, 2021; Li et al., 2020) Social Ski Driver algorithm (Tharwat & Gabel, 2020)

Although *k*-means is one of the most explored algorithms when it comes to clustering, some other methods and algorithms can also be used to solve clustering problems. For example, Kuo et al. (2020) and Nguyen and Kuo (2019) used the Fuzzy *c*-means (FCM) algorithm, which is a clustering algorithm derived from the fuzzy set theory. Thus, in Nguyen and Kuo (2019) an automatic fuzzy clustering using a non-dominated sorting particle swarm optimization algorithm for categorical data is presented. The method can identify the optimal number of clusters based on two objective functions that minimize the global compactness and fuzzy separation that represents the intra-cluster distance and inter-cluster distance. In turn, in Kuo et al. (2020) a metaheuristic-based Possibilistic Multivariate Fuzzy Weighted *c*-means Algorithm (PMFWCM) for clustering mixed (numerical and categorical) data is proposed. In this case, three metaheuristics (GA, PSO, sine cosine algorithm (SCA)) are used in different combinations with the PMFWCM for cluster analysis. Both authors stated that the proposed algorithms work efficiently and determine the optimal number of cluster centers.

5.2.2 Supervised classification methods

Support Vector Machine (SVM) is a popular machine learning methods used for classification tasks. This classifier is guided by a penalty parameter (which determines the trade-off between minimizing the training error and maximizing a classification margin) and kernel parameters (which define the nonlinear transformation of the input feature space to a higher dimensional feature space). Thus, the choice of these parameters is responsible for the classification performance. However, these parameters are empirically selected by trying finite values and keeping the ones that reveal the maximum classification accuracy. This procedure requires an exhaustive search over the search space to find the feasible region and feasible solution, which is a great challenge for SVM (Tharwat & Hassanien, 2018, 2019). According to Wu (2019) the traditional methods to define the kernel parameters by combining the parameter search with *k*-fold cross-validation and grid search have been gradually replaced by the use of an optimization algorithm. Thus, the testing error rate of the machine learning algorithm is minimized by incorporating the optimization algorithm, and the classification performance is improved.

Tharwat and Hassanien (2019) use the Bat Algorithm (BA) to search for the SVM parameters that minimize the testing error rate and improve the classification performance. The proposed algorithm was compared with other classifiers (Grid Search Algorithm (GSA), GA, and PSO) in different benchmark data sets, showing that the proposed model obtained competitive results, that is, test error rates lower than all the other algorithms. At the same time, Zhao et al. (2019) used PSO on a hybrid kernel function support vector machine to solve classification problems. Tharwat and Gabel (2020) proposed a social Ski-driver Algorithm (SSD) that is inspired by different evolutionary optimization algorithms to optimize SVM parameters in unbalanced data sets. In its turn, Dong et al. (2020) searched for the SVM parameters through an improved fireworks algorithm that can adaptively adjust the SVM parameters to get the best combinations in the solution space. The authors He and Fu (2021), Li et al. (2020), and Yu et al. (2021) used PSO to conduct the model training on optimizing the SVM parameters. Wu (2019) used a GA with adaptive genetic operator rates to define the kernel parameters. In all experiments cited, the proposed method has higher accuracy in its classification performance than the conventional (pure) method.

Other interesting approaches involving a hybrid strategy between swarm algorithms and SVM can be seen in the work developed by Moodi et al. (2021). The authors proposed an intelligent adaptive particle swarm optimization-support vector machine that adapts the optimization algorithm parameters, such as the inertia weight and acceleration coefficients. At the same time, Moldovan (2020); Moldovan et al. (2020) used the Horse Optimization Algorithm (HOA) (Moldovan et al., 2020) and the Chicken Swarm Optimization Algorithm (CSOA) (Moldovan, 2020) in different approaches to optimize the regularization parameter and SVM gamma coefficient.

k-**Nearest Neighbors** (*k*-**NN**) is a lazy and non-parametric classification method. This algorithm classifies an object by a majority vote of its *k* neighbors, where *k* is a user-defined parameter (Telikani et al., 2021). However, the classification performance of the *k*-NN algorithm suffers from choosing *k* from a fixed and single value of *k* for all queries in the search stage, and using the simple majority voting rule in the decision stage (Pan et al., 2020). In addition to *k* and the distance function, the importance of neighbor, class, and feature affect the performance of the *k*-NN algorithm. For these reasons, the *k*-NN performance is severely compromised in the unbalanced data set and in the presence of noisy and irrelevant features (Telikani et al., 2021). Thus, Shih and Ting (2019) proposed to optimize the distance function and class-voting weights by a GA on unbalanced data sets. The GA needs to find the optimal feature weights and class weights for *k*-NN. In the evolutionary process, the weights for significant features are expected to be increased, while the irrelevant features and noise are expected to be degraded by shrunken feature weights. A similar approach is presented by Lee et al. (2020), but in this case, the PSO was used to adjust the weight to reflect the importance of features correctly and used the distance judgment strategy to figure out the identical probability of multi-label classification. Both approaches presented higher classification accuracy than other comparable classifiers. In its turn, Jain et al. (2022) take advantage of the capabilities of bio-inspired algorithms, specifically their effective information-sharing mechanisms that enable the algorithm to achieve faster convergence and reduce the likelihood of being trapped in locally optimal solutions to strengthen the performance of *k*-NN classifier. For this, the Grey Wolf Optimization (GWO) algorithm, Chicken Swarm Optimization (CSO) algorithm, and Artificial Bee Colony (ABC) algorithm have been hybridized with the *k*-NN and used in the endeavor to optimize the results of prediction.

Decision Tree (DT) is another very useful algorithm used for classification tasks. A DT consists of a data set partitioned into groups known as nodes. The top node is called the root node, selected using some attribute selection measure or splitting criterion. Under the root node are the internal nodes originating from the division of the data set that constitute the tree branches. At the end of each branch is the terminal node, denoted by leaves, which represent the most appropriate class for the rule (Azevedo et al., 2019). The DT goal is to create a tree model that covers most of the data set and can predict a class by learning simple rules deduced from training data instances (Bida & Aouat, 2021). Several heuristic-based algorithms have been developed to automatically induce DTs and improve the classifier performance, such as Iterative Dichotomiser 3 (ID3), C4.5 Algorithm (C4.5), Classification and Regression Trees (CART), Chi-square Automatic Interaction Detector (CHAID), Quaternion Estimation Algorithm (QUEST). However, these heuristics suffer when exposed to local optima ambush, producing a tree that is not guaranteed to be the global optima (Bida & Aouat, 2021).

To overcome this challenge, some authors combined DT-induced techniques with bio-inspired techniques. Bida and Aouat (2021) described some swarm intelligent DT induction algorithms based on ACO, PSO and BA. Jariyavajee et al. (2019) improved the DT

performance by Joint Approximation Diagonalization of Eigen-matrices algorithm (JADE), which is based on Differential Evolution strategies. Adibi (2019) and Damanik et al. (2019) used GA to optimize DT classifier, whereas Agustina et al. (2019) and Nagra et al. (2020) opted for PSO algorithm to enhance the classifier. Zhou et al. (2020) combined PSO and random forest, an ensemble DT, to classify Ovarian Endometriomas. Comparisons made with classical methods demonstrated that the decision tree method was upgraded to the best prediction method level by incorporating evolutionary and/or swarming techniques.

Other Relevant Hybrid Algorithms have been developed by many authors working on improving bio-inspired algorithms through hybridization.

The original Artificial Bee Colony (ABC) (Karaboga, 2005) is an excellent global optimization algorithm, extremely effective for simple multimodal problems, although it easily suffers from premature convergence in some complex situations (Akay & Karaboga, 2012). The ABC algorithm has a good exploration ability, but its exploitation procedure is not efficient. Hence, Li et al. (2021b) proposed incorporating the modified nearest neighbor in the original ABC to strengthen its optimization capability.

In its turn, Alzaqebah et al. (2021) combined the advantage of the exploration provided by the PSO algorithm and the exploitation ability of the local search method in a feature selection problem. The combination outperformed the original PSO and other comparable approaches, balancing local intensification of the search process and global diversification. Another approach is made by Pravesjit et al. (2021), which combines the PSO with the Evolutionary Rao Algorithm (ERA) (Suyanto et al., 2021) to improve the update position and velocity steps of PSO in a classification problem.

Another interesting approach is proposed by Dixit et al. (2021), which combines a Differential Evolution algorithm with PSO with SVM to detect coronavirus-infected individuals by classifying their chest X-ray images. The proposed method is faster than the pure classifiers DE and PSO, and it was considered a promising approach for classification tasks.

5.2.3 Challenges and future insights in hybrid algorithms

In recent decades, the realm of problem-solving has witnessed the emergence and proliferation of diverse machine learning and evolutionary algorithms. These algorithms have proven to be effective in tackling a wide range of challenges spanning different domains. However, as problems grow in complexity and intricacy, there arises the requirement to merge methodologies and techniques through hybrid methods. Such an integration establishes a robust and powerful framework capable of delivering reliable and efficient solutions swiftly. The future of hybrid methods involving bio-inspired algorithms and machine learning lies in exploring new combinations, improving scalability and efficiency, enhancing explainability, and developing novel algorithms that can tackle complex and large-scale problems across various domains (Pourpanah et al., 2023; Telikani et al., 2021). However, several critical issues persist that have not yet received sufficient research attention. Some of these significant research gaps are summarized below.

Evolutionary deep learning: the combination of evolutionary search algorithms and deep neural networks enables the automatic design of network architectures, feature selection, and hyperparameter optimization (Kazadi Mbamba & Batstone, 2023; Sulaiman et al., 2023). Future directions may involve the development of hybrid algorithms that incorporate genetic operators with deep learning models to enhance optimization and adaptation capabilities.

Neuroevolution: it refers to using evolutionary algorithms to evolve artificial neural networks. Future directions include exploring new algorithms that combine swarm and evolutionary algorithms, neural networks, and reinforcement learning to improve the efficiency and scalability of neuroevolution techniques (Martinez et al., 2021).

Memetic computing: it combines evolutionary and local search algorithms to benefit from their strengths, enabling global and local exploitation. Future advancements may involve integrating memetic computing with bio-inspired and deep learning algorithms to effectively handle complex, high-dimensional data (Huang et al., 2019; Khalfi et al., 2023; Yu et al., 2023).

Hybrid methods comparison: most publications have focused on comparing bio-inspired or hybrid methods with traditional ones through mathematical analysis of runtime, convergence guarantee, and parameter configurations. Few of these studies systematically compared the performance of different bio-inspired algorithms in machine learning tasks or different machine learning techniques in bio-inspired optimization algorithms. This leads to a lack of experimental results to select the most suitable method for a particular combination. The unavailability of such surveys may be due to the lack of publicly available source code, variation of encoding techniques, different objective functions, and evolutionary operators. As a result, there is a vast amount of published work since numerous metaheuristic algorithms can be combined with machine learning. Still, there is immense difficulty in pointing out which combinations are most appropriate or even why one is more advantageous than the other.

Multi/Many-objective approaches: while most hybridization research between optimization and machine learning focuses on single-objective approaches, real-world problems often involve multiple objectives (Kang et al., 2023). Besides, existing multi-objective algorithms such as NSGA-II, PESA-II, and SPEA-II face challenges with more than four objectives (Telikani et al., 2021). Thus, exploring the combination of machine learning in multi/many-objective could be a valuable and fruitful source of innovation. Moreover, in the same scope, techniques for choosing the Pareto front solution can be explored through inspiration in machine learning techniques (Wang et al., 2023).

Big data approaches: it offers new opportunities for algorithms research, but it also brings challenges such as computational costs, huge high-dimensional sample sizes, storage impasse, and error extent. So, hybrid methods can be a way to attack this kind of problem.

Collaborative algorithms: the transition from Industry 4.0 to 5.0 is marked by several dominant trends (sustainability, customization, real-time decision-making, and the constant transformation of the market). One research avenue involves improving optimization algorithms and machine learning techniques to make them more globally applicable and accurate. Besides, implementing combined strategies using multiple algorithms that cooperate to solve a problem can lead to identifying the most suitable solution.

Integration of human expertise: Human-in-the-loop approaches will play a vital role in decision-making, problem-solving, and creativity, combining the strengths of both human intelligence and machine learning algorithms. This integration enables human experts to provide guidance, validate results, and incorporate domain knowledge into the hybrid models (Bailey et al., 2023; Li et al., 2023).

Explainability and interpretability: as hybrid algorithms become more sophisticated, there is a growing demand for explainability and interpretability. Future trends involve the development of hybrid models that provide transparent explanations for their decisions. This enables users to understand and trust the outputs of the algorithms, particularly in domains where interpretability is crucial, such as healthcare, finance, and law.

The future of hybrid algorithms lies in their integration with deep learning, increased explainability, meta-learning, cross-domain applications, real-time adaptation, and integration of human expertise. These trends aim to improve hybrid algorithms' performance, efficiency, and versatility in tackling complex real-world problems.

6 SWOT analysis and research questions answers

The performance of an algorithm depends on several factors, as the solution quality and consumed budget are the most significant measures in the algorithm performance assessment. As previously mentioned by the No-Free-Lunch theorem, there is no universal and perfect method and/or algorithm to solve all optimization problems (Wolpert & Macready, 1997). Based on this, it is a hard task to define the best strategy (or algorithm) without considering the problem and the data information, leaving the option of analyzing the characteristics of those we have at our disposal and finding ways to choose the most appropriate one.

6.1 SWOT analysis

In this section, a SWOT (Strengths, Weaknesses, Opportunities, and Threats) analysis is performed (Gürel & Tat, 2017). This analysis consists of evaluating the characteristics of the algorithm and developing strategies to assist in choosing the most appropriate algorithm to solve a specific mathematical model.

First, the 10 most cited algorithms in the keywords of the papers in the literature review database were identified. Table 2 shows the result of the search, according to the name of the algorithm, Type of Method (Optimization or Machine Learning), and the number of times that the algorithm was cited in the keywords.

Thus, the 10 algorithms presented were analyzed according to the SWOT criterion, as shown in Table 3, where the first column defines the name of the algorithm, then the following four columns present the SWOT parameters (Strengths, Weaknesses, Opportunities, and Threats) identified for each algorithm, and in the last column there is the indication of some works that combined the algorithm presented in the first column with another approach, that is, some examples of hybrid methods.

Table 2 Most cited algorithms

Algorithm	Type of method	Citations
Particle swarm optimization	Optimization	137
Support vector machine	Machine Learning	88
Genetic algorithm	Optimization	71
k-means	Machine learning	31
k-nearest neighbor	Machine learning	29
Differential evolution	Optimization	28
Decision trees	Machine learning	28
Artificial bee colony	Optimization	23
Random forest	Machine learning	19
Ant colony optimization	Optimization	18

Table 3 SWOT analysis of the 10 most cited algorithms

Algorithm	Strengths	Weaknesses	Opportunities	Threats	Hybrid approaches
Particle swarm optimization	High exploration ability. Easily parallelized. Simple and easy to implement	Poor local search ability. Poor performance into high-dimensional models. Dependent of initial parameters estimation. Low convergence rate in the iterative process	Explore initial parameters choice. Improve position and speed update. Improve the ability to escape from local optimum. Explore convergence speed controller and memory requirement	Algorithm with high global search ability. Deterministic methods	Tharwat and Gabel (2020); Pravesjit et al. (2021) Chai et al. (2021); Willis et al. (2020) Ratanavilisagul (2020); Garcia-Nieto et al. (2019) Pham et al. (2020)
Support vector machine	Works well when there is a clear margin of separation between classes. Effective in high-dimensional space. Performs linear and non-linear classifications. Solves classification and regression problems	Not suitable for large data sets. Sensitive to noise data. High memory consumption on large volumes of data	Improve parameters adaptation (inertia weight and acceleration coefficients). Minimize test error rate and maximize the hit margin	Algorithms not dependent of initial parameters choice. Deterministic methods	Wu (2019); Tharwat and Hassanien (2019) Yu et al. (2021); Moodi et al. (2021) Moodi et al. (2020); Al-Zoubi et al. (2021)
Genetic algorithm	High exploration ability. Easily parallelized. One of the most famous and explored bio-inspired algorithm	Expensive to implement and can be time-consuming. Can get stuck in local optima. Hyperparameter setting. High computational complexity. Dependent on choice of genetic operator rates	Explore strategies to genetic operator rates estimation. Study genetic operator rate based on the search stage. Improve exploration and exploitation process. Explore parallelization strategies	Faster algorithms. Algorithm not dependent on operator rates choice. Simple coding algorithm. Deterministic methods	Wu (2019); AL-Behadili et al. (2020) Abualigah and Dulaimi (2021); Li et al. (2021a) Essiet et al. (2019); Cherki et al. (2019)

Table 3 (continued)

Algorithm	Strengths	Weaknesses	Opportunities	Threats	Hybrid approaches
k- means clustering algorithm	One of the most famous and explored clustering algorithm. Speediness. Simple and easy to implement. Easily adapts to new examples	Can get stuck in local optima. Dependent on initial parameter estimation. Sensitive to sizes and density data sets. Sensitive to noises and outliers data. Do not work with categorical data	Explore strategies to automatically definition of *k* value. Explore strategies to avoid local optimal stopping. Improve search performance	Flexibility and adaptability algorithms, such as fuzzy clustering methods. Automatic clustering approaches (do not dependent on initial choices). Deterministic methods	Zhou et al. (2019); Chen et al. (2020) El-Shorbagy et al. (2019); Ratanavilisagul (2020) Thalamala et al. (2020)
k- Nearest neighbors	No training period required (instance based learning). Much faster than other algorithms that require training. Simple and easy to implement. New data can be added seamlessly without affecting the accuracy of the algorithm	Dependent on initial parameters estimation. It is severely compromised in the unbalanced data set. Sensitive in noisy data, irrelevant features and missing values. Does not work well with high dimensions nor large data sets	Explore strategies to work with unbalanced data and noise data. Explore strategies to automatically definition of *k* value. Explore adaptive strategies to deal with the *k* value	Flexible algorithms in terms of *k* value adaptation. Deterministic methods	Pan et al. (2020); Shih and Ting (2019) Li et al. (2021b); Gao et al. (2019) Huang et al. (2019); Jain et al. (2022)
Differential evolution	Simple and efficiency. Local searching property. Speediness	Dependent on choice of genetic operator rates. Poor exploitation ability. Low convergence velocity. Sensitive to the choice of the control parameters. Not suitable for large data sets. Poor performance in complex problems	Study modification in the genetic operators. Adjust control parameters and/or introduce new operators	Algorithm not dependent of initial parameters choice or the ones that do it automatically. Deterministic methods	Wang et al. (2019); Deng et al. (2021) Li et al. (2020); Ahmad et al. (2022)

Table 3 (continued)

Algorithm	Strengths	Weaknesses	Opportunities	Threats	Hybrid approaches
Decision tree algorithms	Performs classification and regression problems. Captures nonlinear relationships. Easy to understand and interpret. Results graphic visualization. Useful in data exploration. Less data preparation needed. Non-parametric algorithm	Requires pruning strategy to increase the performance. Not suitable for large data sets. High complexity and overfitting on large data sets	Explore evolutionary pruning methods. Explore split and merge strategies based on swarm intelligence. Minimize the error rate	Flexibility algorithms (fuzzy decisions trees). Algorithms capable of dealing with outliers, missing data and overfitting problems. Deterministic methods	Barros et al. (2012); Agustina et al. (2019) Bida and Aouat (2021); Jariyavajee et al. (2019) Adibi (2019); Damanik et al. (2019) Nagra et al. (2020)
Artificial bee colony	Very efficient for basic multimodal problems. Flexible and robust. High exploration ability. Simple and easy to implement. Handle complex functions	Slow when in sequential processing. Can get stuck in local optima on complex problems. Poor exploitation ability	Improve exploitation procedure. Combine strategies to balance the exploration and exploitation during the search process. Explore modification in the genetic operators	Deterministic methods	Karaboga et al. (2020); Li et al. (2021b) Ozger et al. (2019); Wang et al. (2020)

Table 3 (continued)

Algorithm	Strengths	Weaknesses	Opportunities	Threats	Hybrid approaches
Random forest	Few prone to overfitting. Solves classification and regression problems. Works categorical and continuous variables. Handles missing values. No feature scaling is required. Handles non-linear parameters efficiently. Very stable. Little affected by noise data	Complexity. Time consuming. Longer training period. Difficulties working with unbalanced data sets.	Improve voting strategy based on weighting data mining information (data reliability). Explore strategies to speed data processing	Faster algorithms. Algorithms able to work with unbalanced data sets. Deterministic methods	Bi et al. (2020); Farrell et al. (2020) Mittal et al. (2021); Asadi et al. (2021) Moldovan et al. (2019)
Ant colony optimization	Positive feedback leads to rapid discovery of good solutions. Attractive for searching a large feature space	Coding is not straightforward. Can get struck in local optima	Introduce and/or modify the genetic operators. Introduce elitist ants and weakened strategy to control the pheromone impact. Explore the random factor introduction to modify the search direction.	Deterministic methods	AL-Behadili et al. (2020); Montemayor (2019) Deng et al. (2019); Gao (2021)

6.2 Research questions answers

As seen in previous sections, several researchers are working on hybrid strategies to highlight the algorithm capabilities. One of the main objectives of this literature review is to explore the characteristics of the existing approaches and also of the hybrid methods that are being developed recently. Thus, based on an extensive literature analysis, the research questions defined in the beginning can be answered as follows.

- **RQ1: What are the difficulties of classical bio-inspired optimization algorithms and machine learning algorithms?** Since we are faced with a wide variety of algorithms, whether optimization or machine learning, that use different strategies to perform their functions, pointing out the difficulties of each one is not a viable task. In general, based on the literature, there are two points constantly mentioned in the works that concern the main difficulties encountered by the so-called pure algorithms; the first refers to premature convergence to the local optimum (Li et al., 2021b; Pacifico & Ludermir, 2021) and the second point refers to parameter estimation (Dong et al., 2020; Lee et al., 2020; Moldovan, 2020; Shih & Ting, 2019; Wu, 2019). Many authors report that machine learning algorithms suffer from premature convergence, so they employ swarm-based optimization algorithms to avoid local optima. Moreover, within bio-inspired techniques, strong attention is directed to the PSO algorithm due to its simplicity and speed of convergence. However, other lesser-known bio-inspired algorithms can also be used for the same purposes, leading to results as good as PSO. On the other hand, regarding parameter estimation, it is clear that there is a large community exploring ways to automate the estimation of these parameters. Regarding clustering methods, the k-means algorithm is the most mentioned when the task is parameters estimation. The same occurs for parameter estimation of the SVM and k-NN classifiers. The k-means, SVM, and k-NN classifiers are heavily dependent on these initial choices; in addition, a given parameter value may result in a good performance for one classification problem and failure for another. This becomes the choice of parameter a big challenge, mainly when there is no *a priori* information about the problem. Furthermore, algorithms that use genetic operators such as PSO, GA, DE, ABC, and ACO require the estimation of the genetic operator rates, which is also a challenge in the search for an optimal value. Notwithstanding other difficulties, the particularities of each algorithm can be verified in the Weaknesses column of Table 3.
- **RQ2: What are the methods or techniques already developed that combine optimization and machine learning in order to improve algorithms performance?** Throughout this work, several research were presented that, in their particularities, help to answer this question. Among these works, the strong presence of strategies that used the PSO algorithm stands out, due to its speed and robustness, and exploration capabilities, to strengthen both optimization and machine learning algorithms; thus, it is the most mentioned algorithm in the paper's keywords of the data set. Due to the large number of works found and their peculiarities, it is a hard task to list all combinations of algorithms found in this work. However, more details can be seen in Sect. 5 as well as in Table 3. In general, the items listed in the Opportunities column of Table 3, refer to points that are being worked on in the cited references, such as methods and techniques that are being developed to improve the algorithm performance. In this sense, it is worth highlighting the estimation of initial parameters

Atabay et al. (2016), El-Shorbagy et al. (2019) and Wu (2019) and the strengthening of exploitation strategies to avoid getting stuck in local optima (Li et al., 2021b; Pacifico & Ludermir, 2021). Furthermore, the use of optimization algorithms to minimize the classification error in machine learning algorithms (Agustina et al., 2019; Nagra et al., 2020; Wu, 2019). Therefore, in the mentioned references, it is possible to find different combinations of algorithms and techniques that propose to solve the weaknesses mentioned in Table 3.

- **RQ3: What are the potentialities of the optimization and learning algorithm developed?** The use of hybrid tools allows some of the difficulties encountered by the methods to be mitigated or eliminated. Therefore, the results that have been obtained with hybridization are very promising. Most of the time, hybrid algorithms outperform pure algorithms in several aspects, such as speed, exploration and/or exploitation abilities, accuracy, among other aspects, without requiring more computational resources than pure methods. Furthermore, the estimation of parameters, one of the most mentioned obstacles, which is done empirically in pure methods, is now based on knowledge of the data, which provides a result that is many times superior and more appropriate for the learning task, free from any possible bias. In short, the potential of the developed hybrid methods refer to the items listed in the Opportunities column of the Table 3, which are used in the solution of the identified weakness point, listed in the Weaknesses column of the same table.

7 Conclusions

In this paper, an extensive overview and bibliometric analysis of the literature on hybrid approaches involving optimization and machine learning algorithms were developed. Thus, a theoretical foundation on optimization and then machine learning was presented. After that, the results of the literature review are presented. Initially, a historical survey of works published since 1970 was carried out, involving the main themes contemplated in this work: Optimization, Machine Learning, Swarm and Evolutionary algorithms, in order to analyze the evolution of the theme throughout the years. To generate the data set of works to be analyzed a logic search was defined and applied to three databases (Scopus, IEEE, and WoS) together with some restrictions of year, language, and publisher type. Thereby, 1007 articles published between 2019 and 2021, provided by three databases (479 from Scopus, 200 from IEEE, and 450 WoS) were selected and analyzed in a systematic and bibliometric way, which resulted in an extensive range of works. Finally, a SWOT analysis of the most frequent algorithms in the data set was performed to identify the strengths, weaknesses, opportunities and threats of each one and point out some works that explore such aspects and that somehow developed a hybrid approach. Furthermore, some examples of hybrid approaches involving the algorithms considered in the SWOT analysis were presented.

While the majority of literature review work focuses on specific algorithms classes of algorithms (Ahmad et al., 2022; Barros et al., 2012; Ezugwu et al., 2021; Karaboga et al., 2020; Shami et al., 2022), this work aimed to be more embracing. Literature review works involving hybrid methodologies that combine optimization and machine learning are rare to be found due to the extensive nature of both methodologies and the challenges in conducting a comprehensive examination of existing techniques. This work stands out from others due to its comprehensive methodological approach, encompassing both optimization

and machine learning algorithms used innovatively through hybrid algorithms. Additionally, the study considers different methods, applies logical restrictions for clear scope and focus on the most relevant aspects, conducts an in-depth critical evaluation of the reviewed studies, and emphasizes the practical applicability of the developed hybrid approaches. These characteristics provide a unique perspective, offering comprehensive and targeted analysis and valuable insights for advancing the field.

Over the past half-decade, a notable surge in the advancement of bio-inspired algorithms has been observed, accompanied by a compelling inclination towards the fusion of diverse methodologies. Thereby, the present study has unveiled emerging points within the realm of hybrid techniques encompassing optimization and machine learning, thereby shedding light on prospective avenues for further exploration. Through a deep examination of the literature, it is discernible that there exists substantial scope for delving into bio-inspired strategies, particularly those rooted in swarm intelligence (Kang et al., 2023; Martinez et al., 2021; Yu et al., 2023). These techniques have demonstrated impressive results in optimizing problems and algorithms, and they possess the potential to enhance machine learning algorithms.

Through the systematic review, valuable contributions have been made to various hybrid approaches encompassing optimization and machine learning techniques. Each algorithm presented in this context exhibits its own set of strengths and weaknesses, requiring careful assessment to determine the most suitable methods for addressing a given problem. Among the multitude of aspects explored within these algorithms, two stand out: parameter estimation and search technique improvement, both crucial to avoid local optima. The estimation of initial parameters made empirically is seen as a weakness, given the subjectivity of the choice and possible bias of the decision maker, which is capable of compromising algorithm performance. Similarly, being stuck in local optima proves highly undesirable in global optimization algorithms, as it yields error-inducing solutions. On the other hand, the opportunities and threats within this domain are intricately intertwined with the identified strengths and weaknesses, as exploring these aspects enables the mitigation of weaknesses through the potential offered by other methods, i.e. hybrid approaches. Furthermore, the threats posed by a particular algorithm can be effectively addressed by the strengths of alternative algorithms, as elucidated in the No-Free-Lunch theorem (Wolpert & Macready, 1997).

Considering the relevance of the hybrid methods in solving real-world problems, numerous unexplored avenues remain for leveraging optimization and machine learning tasks. Looking toward future advancements and literature exploration, it would be compelling to delve into variations of existing hybrid approaches. For example, investigating whether a hybridization approach that yields superior outcomes for classification tasks also translates into equivalent improvements when applied to regression tasks would be of great interest. This comparative analysis of the performance and adaptability of hybrid techniques across different problem domains holds the potential to unearth valuable insights for further development. In conclusion, considering the large volume of literature available in the area, it is possible that we have left out some important contributions due to the hard restriction applied in the logic search. Thereby, in future research, it is recommended to include the restriction applied in the logic search, as it is an exploration of the literature review for each method, i.e, supervised methods, unsupervised methods reinforcement methods, evolutionary algorithms and swarm algorithms; in this way, it would be possible to expand the existing literature review and provide a more constructive discussion of the benefits and drawbacks of the state-of-the-art.

Availability of data and materials the datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Code availability The algorithms applied are available from the corresponding author on reasonable request.

Declarations

Conflict of interest Not applicable.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

References

Abarghouei, A., Ghanizadeh, A., Sinaie, S., & Shamsuddin, S. (2009). A survey of pattern recognition applications in cancer diagnosis. In *International conference of soft computing and pattern recognition* (pp. 448–453).

Absara, A., Kumar, S., Lenin Fred, A., Ajay Kumar, H., & Suresh, V. (2020). An improved fuzzy clustering segmentation algorithm based on animal behavior global optimization. *Advances in Intelligent Systems and Computing, 1048*, 737–748.

Abualigah, L., & Dulaimi, A. (2021). A novel feature selection method for data mining tasks using hybrid sine cosine algorithm and genetic algorithm. *Cluster Computing, 24*, 2161–2176.

Adibi, M. (2019). Single and multiple outputs decision tree classification using bi-level discrete-continues genetic algorithm. *Pattern Recognition Letters, 128*, 190–196.

Agrusti, F., Bonavolontà, G., & Mezzini, M. (2019). University dropout prediction through educational data mining techniques: A systematic review. *Journal of E-Learning and Knowledge Society, 15*, 161–182.

Agustina, C., Purwati, N., Sulistyo, G., Hasan, N., & Rapiyanta, P. (2019). Decision tree algorithm using particle swarm optimization to improve the accuracy of detection malnutrition in toddler. In *Proceedings of the 1st international conference on science and technology for an internet of things, ICSTI-Indonesia* (pp. 1–11).

Ahmad, M., Isa, N., Lim, W., & Ang, K. (2022). Differential evolution: A recent review based on state-of-the-art works. *Alexandria Engineering Journal, 61*, 3831–3872.

Akay, B., & Karaboga, D. (2012). A modified artificial bee colony algorithm for real-parameter optimization. *Information Sciences, 192*, 120–142.

AL-Behadili, H., Ku-Mahamud, K., & Sagban, R. (2020). Hybrid ant colony optimization and genetic algorithm for rule induction. *Journal of Computer Science, 16*, 1019–1028.

Al-Zoubi, A., Hassonah, M., Heidari, A., Faris, H., Mafarja, M., & Aljarah, I. (2021). Evolutionary competitive swarm exploring optimal support vector machines and feature weighting. *Soft Computing, 25*, 3335–3352.

Albarakati, N., & Obradovic, Z. (2019). Multi-domain and multi-view networks model for clustering hospital admissions from the emergency department. *International Journal of Data Science and Analytics, 8*, 385–403.

Alrosan, A. (2021). Automatic data clustering based mean best artificial bee colony algorithm. *Computers, Materials and Continua, 68*, 1575–1593.

Alves, F., Pereira, A., Fernandes, A., & Leitão, P. (2018). Optimization of home care visits schedule by genetic algorithm. In *Bioinspired optimization methods and their applications, proceedings of 8th international conference on* (pp. 1–12).

Alzaqebah, M., Jawarneh, S., Mohammad, R., Alsmadi, M., Al-Marashdeh, I., Ahmed, E., Alrefai, N., & Alghamdi, F. (2021). Hybrid feature selection method based on particle swarm optimization and adaptive local search method. *International Journal of Electrical and Computer Engineering, 11*, 2414–2422.

Aria, M., & Cuccurullo, C. (2017). Bibliometrix: An r-tool for comprehensive science mapping analysis. *Journal of Informetrics, 11*, 959–975.

Asadi, S., Roshan, S., & Kattan, M. (2021). Random forest swarm optimization-based for heart diseases diagnosis. *Journal Of Biomedical Informatics, 115*, 103690.

Aslam, N., Khan, I., Alkhalifah, S., Al-Sadiq, S., Bughararah, S., Al-Otabi, M., & Al-Odinie, Z. (2021). Predicting diabetic patient hospital readmission using optimized random forest and firefly evolutionary algorithm. *Engineering And Information Technology: International Journal On Advanced Science*.

Atabay, H., Sheikhzadeh, M., & Torshizi, M. (2016). A clustering algorithm based on integration of k-means and pso. In *1st conference on swarm intelligence and evolutionary computation (CSIEC 2016)-higher education complex of bam* (pp. 59–63).

Azevedo, B. (2020). Study of genetic algorithms for optimization problems. Master's thesis, Instituto Politecnico de Braganca Escola Superior de Tecnologia e Gestao, Portugal.

Azevedo, B., Amoura, Y., Rocha, A., Fernandes, F., Pacheco, M., & Pereira, A. (2022). Analyzing the mathe platform through clustering algorithms. In *Computational science and its applications-ICCSA 2022 workshops* (pp. 201–218).

Azevedo, B., Bressan, G., Agulhari, C., Santos, H., & Endo, W. (2019). Three-phase induction motors faults classification using audio signals and decision trees. *Applied Mathematics and Information Sciences, 13*, 847–858.

Azevedo, B., Brito, T., Lima, J., & Pereira, A. (2021). Optimum sensors allocation for a forest fires monitoring system. *Forests 12*.

Bailey, W., Goldsmith, J., & Harrison, B., Others (2023). Fast approximate bi-objective pareto sets with quality bounds. *Autonomous Agents And Multi-Agent Systems, 37*.

Bansal, J., Singh, P., & Nikhil, R. (2019). Evolutionary and swarm intelligence algorithms. Studies In *Computational intelligence*.

Barros, R., Basgalupp, M., Carvalho, A., & Freitas, A. (2012). A survey of evolutionary algorithms for decision-tree induction. *IEEE Transactions On Systems, Man, And Cybernetics , Part C (Applications And Reviews), 42*, 291–312.

Bi, X., Hu, X., Wu, H., & Wang, Y. (2020). Multimodal data analysis of Alzheimer's disease based on clustering evolutionary random forest. *IEEE Journal of Biomedical and Health Informatics, 24*, 2973–2983.

Bida, I., & Aouat, S. (2021). Swarm intelligence-based decision trees induction for classification—a brief analysis. In *Proceeding of 2nd international workshop on human-centric smart environments for health and well-being (IHSH)* (pp. 165–170).

Bishop, C. (2006). *Pattern recognition and machine learning*. USA: Springer.

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: From natural to artificial systems*. Oxford University Press.

Boyd, S. (2004). *Convex optimization*. Cambridge University Press.

Bressan, G., & Azevedo, B. (2017). Classificação de gêneros musicais latinos e suas emoções: Abordagens bayesiana e fuzzy. *Tendências Em Matemática Aplicada E Computacional, 18*, 369–384.

Bressan, G., Azevedo, B., Santos, H., Endo, W., Agulhari, C., Goedtel, A., & Scalassara, P. (2021). Bayesian approach to infer types of faults on electrical machines from acoustic signal. *Applied Mathematics and Information Sciences, 15*, 353.

Buenaño-Fernandez, D., Villegas-CH, W., & Luján-Mora, S. (2019). The use of tools of data mining to decision making in engineering education—A systematic mapping study. *Computer Applications in Engineering Education, 27*, 744–758.

Chai, Z., Nwachukwu, A., Zagayevskiy, Y., Amini, S., & Madasu, S. (2021). An integrated closed-loop solution to assisted history matching and field optimization with machine learning techniques. *Journal of Petroleum Science and Engineering, 198*, 108204.

Chen, J., Qi, X., Chen, L., Chen, F., & Cheng, G. (2020). Quantum-inspired ant lion optimized hybrid k-means for cluster analysis and intrusion detection. *Knowledge-Based Systems, 203*, 106167.

Cherif, W. (2018). Optimization of k-nn algorithm by clustering and reliability coefficients: Application to breast-cancer diagnosis. In *Proceedings of the first international conference on intelligent computing in science - ICDS* (Vol. 127, pp. 293–299).

Cherki, I., Chaker, A., Djidar, Z., Khalfallah, N., & Benzergua, F. (2019). A sequential hybridization of genetic algorithm and particle swarm optimization for the optimal reactive power flow. *Sustainability, 11*.

Cicceri, G., Inserra, G., & Limosani, M. (2020). A machine learning approach to forecast economic recessions-an Italian case study. *Mathematics*.

Cichosz, P. (2015). *Data mining algorithms explained using R*. John Wiley & Sons.

Čorić, R., Dumić, M., & Jakobović, D. (2017). Complexity comparison of integer programming and genetic algorithms for resource constrained scheduling problems. In *Proceeding Of 40th international convention on information and communication technology, electronics and microelectronics (MIPRO)* (pp. 1182–1188).

Cotta, C., Mathieson, L., & Moscato, P. (2018) . Memetic algorithm. In Martí, P., R., Pardalos, P. M., Resende, M. G. C. (Eds.), *Handbook of heuristics*.

Damanik, I., Windarto, A., Wanto, A., Poningsih, Andani, S., & Saputra, W. (2019). Decision tree optimization in c4.5 algorithm using genetic algorithm. *Journal Of Physics: Conference Series*, *1255*.

Deng, W., Shang, S., Cai, X., et al. (2021). An improved differential evolution algorithm and its application in optimization problem. *Soft Computing, 25*, 5277–5298.

Deng, W., Xu, J., & Zhao, H. (2019). An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem. *IEEE Access, 7*, 20281–20292.

Deshpande, A., & Triantaphyllou, E. (1998). A greedy randomized adaptive search procedure (grasp) for inferring logical clauses from examples in polynomial time and some extensions. *Mathematical And Computer Modelling, 27*, 75–99.

Dewey, A., & Drahota, A. (2016). Introduction to systematic reviews: Online learning module cochrane training. https://training.cochrane.org/interactivelearning/module-1-introduction-conducting-syste matic-reviews. Accessed on 2022 September 01, 2022

Dhal, K., Gálvez, J., & Das, S. (2020). Toward the modification of flower pollination algorithm in clustering-based image segmentation. *Neural Computing and Applications, 32*, 3059–3077.

Dixit, A., Mani, A., & Bansal, R. (2021). Cov2-detect-net: Design of Covid-19 prediction model based on hybrid de-pso with svm using chest x-ray images. *Information Sciences, 571*, 676–692.

Dong, D., Ye, Z., Su, J., Xie, S., Cao, Y., & Kochan, R. (2020). A malware detection method based on improved fireworks algorithm and support vector machine. In *2020 IEEE 15th international conference on advanced trends in radioelectronics, telecommunications and computer engineering (TCSET)* (pp. 846–851).

Dorigo, M., & Stützle, T. (2003). The ant colony optimization metaheuristic: Algorithms, applications, and advances. In *Handbook of metaheuristics* (pp. 250–285).

Dutta, D., Sil, J., & Dutta, P. (2019). Automatic clustering by multi-objective genetic algorithm with numeric and categorical features. *Expert Systems With Applications, 137*, 357–379.

Edwin Dhas, P., & Sankara Gomathi, B. (2020). A novel clustering algorithm by clubbing ghfcm and gwo for microarray gene data. *Journal of Supercomputing, 76*, 5679–5693.

Eesa, A., Orman, Z. (2020). A new clustering method based on the bio-inspired cuttlefish optimization algorithm. *Expert Systems*, *37*.

El-Shorbagy, M., Ayoub, A., Mousa, A., & El-Desoky, I. (2019). An enhanced genetic algorithm with new mutation for cluster analysis. *Computational Statistic, 34*, 1355–1392.

Essiet, I., Sun, Y., & Wang, Z. (2019). Improved genetic algorithm based on particle swarm optimization-inspired reference point placement. *Engineering Optimization, 51*, 1097–1114.

Estivill-Castro, V., & Yang, J. (2000). Fast and robust general purpose clustering algorithms. In *Proceedings Of The 6th Pacific RIM international conference on artificial intelligence* (pp. 208–218).

Ezugwu, A., Agbaje, M., Nahla, A., Els, R., Chiroma, H., & Elaziz, M. (2020). A comparative performance study of hybrid firefly algorithms for automatic data clustering. *IEEE Access, 8*, 121089–121118.

Ezugwu, A., Shukla, A., & Agbaje, M. (2021). Others: Automatic clustering algorithms: A systematic review and bibliometric analysis of relevant literature. *Neural Computing and Applications, 33*, 6247–6306.

Farrell, M., Ramadhani, K., & Suyanto, S. (2020). Combined firefly algorithm-random forest to classify autistic spectrum disorders. In *2020 3rd international seminar on research of information technology and intelligent systems (ISRITI)* (pp. 505–508).

Fera, M., Fruggiero, F., Lambiase, A., Macchiaroli, R., & Todisco, V. (2018). A modified genetic algorithm for time and cost optimization of an additive manufacturing single-machine scheduling. *International Journal of Industrial Engineering Computations, 9*, 423–438.

Fávero, P., & Zoucas, F. (2016). Redes neurais para previsão da produção industrial de diferentes segmentos. *Produto E Produção, 17*(2), 53–70.

Fürnkranz, J., Gamberger, D., & Lavrač, N. (2012). *Foundations of rule learning*. Springer.

Galatzer-Levy, I., Ruggles, K., & Chen, Z. (2018). Data science in the research domain criteria era: Relevance of machine learning to the study of stress pathology, recovery, and resilience. *Chronic Stress, 2*, 2470547017747553.

Gandomi, A., Yang, X., Talatahari, S., & Alavi, A. (2013). *Metaheuristic applications in structures and infrastructures*. Elsevier.

Gao, W. (2021). Modified ant colony optimization with improved tour construction and pheromone updating strategies for traveling salesman problem. *Soft Computing, 25*, 3263–3289.

Gao, J., Dong, M., Jing, C. (2019). A cooperative coevolutionary algorithm for knn training set optimization. In *Association for Computing Machinery*.

Garcia-Nieto, J., Lopez-Camacho, E., Garcia-Godoy, M., Nebro, A., & Aldana-Montes, J. (2019). Multi-objective ligand-protein docking with particle swarm optimizers. *Swarm and Evolutionary Computation, 44*, 439–452.

Giuzio, M. (2017). Genetic algorithm versus classical methods in sparse index tracking. *Decisions in Economics and Finance, 40*, 243–256.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research, 13*, 533–549.

Goldberg, D. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Professional.

Gopal, M. (2018). *Applied machine learning*. McGraw Hill Education Private Limited.

Gürel, S., & Tat, M. (2017). Swot analysis: A theoretical review. *The Journal of International Social Research, 10*, 994–1006.

Han, J., Kamber, M., & Pei, J. (2012). *Data mining: Concepts and techniques*. Morgan Kaufmann Publishers.

He, D., & Fu, J. (2021). Application of svm based on improved particle swarm optimization algorithm in epileptic seizure detection. In *Proceeding of 40th Chinese control conference* (Vol. 2021, pp. 7082–7087).

Holland, J. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press.

Houssein, E., Ahmad, M., Hosney, M., & Mazzara, M. (2021). Classification approach for covid-19 gene based on harris hawks optimization. *Studies in Systems, Decision and Control, 358*, 575–594.

Hua, Y., Liu, Q., Hao, K., & Jin, Y. (2021). A survey of evolutionary algorithms for multi-objective optimization problems with irregular pareto fronts. *IEEE/CAA Journal of Automatica Sinica, 8*, 303–318.

Huang, K., Wen, Y., & Ting, C. (2019). Enhancing k-nearest neighbors through learning transformation functions by genetic programming. In *2019 IEEE congress on evolutionary computation (CEC)* (pp. 1891–1897).

Huang, K., Wu, Z., Peng, H., Tsai, M., Hung, Y., & Lu, Y. (2019). Memetic particle gravitation optimization algorithm for solving clustering problems. *IEEE Access, 7*, 80950–80968.

Iglesias, F., Zseby, T., & Zimek, A. (2021). Clustering refinement. *International Journal of Data Science and Analytics, 12*, 333–353.

Jacob, C. (2001). *Illustrating Evolutionary Computation with Mathematica*. Berlin: Morgan Kaufmann.

Jain, R., Batra, J., & Kar, A. (2022). A hybrid bio-inspired computing approach for buzz detection in social media. *Evolutionary Intelligence, 15*, 349–367.

Jariyavajee, C., Polvichai, J., & Sirinaovakul, B. (2019). Searching for splitting criteria in multivariate decision tree using adapted jade optimization algorithm. In *2019 IEEE symposium series on computational intelligence (SSCI)* (pp. 2534–2540).

Jones, D., & Martins, J. (2021). The direct algorithm: 25 years later. *Journal of Global Optimization, 79*, 521–566.

Jones, D., Perttunen, C., & Stuckman, B. (1993). Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Applications, 79*, 157–181.

Kang, Y., Xie, W., Wang, X., Wang, H., Wang, X., & Li, J. (2023). Mopisde: A collaborative multi-objective information-sharing de algorithm for software clustering. *Expert Systems with Applications, 226*, 120207.

Karaboga, D. (2005). An idea Based on Honey Bee Swarm for Numerical Optimization. Ph.D. thesis, Erciyes University, Engineering Faculty, Computer Engineering Department .

Karaboga, D., Akay, B., & Karaboga, N. (2020). A survey on the studies employing machine learning (ml) for enhancing artificial bee colony (abc) optimization algorithm. *Cogent Engineering, 7*, 1855741.

Kazadi Mbamba, C., & Batstone, D. (2023). Optimization of deep learning models for forecasting performance in the water industry using genetic algorithms. *Computers and Chemical Engineering, 175*, 108274.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks* (Vol. 4, pp. 1942–1948).

Khalfi, S., Iacca, G., & Draa, A. (2023). A single-solution-compact hybrid algorithm for continuous optimization. *Memetic Computing, 15*, 155–204.

Khattar, N., Sidhu, J., & Singh, J. (2019). Toward energy-efficient cloud computing: a survey of dynamic power management and heuristics-based optimization techniques. *The Journal Of Supercomputing, 75*, 4750–4810.

Kirkpatrick, S., Gelatt, C., & Vecchi, M. (1983). Optimization by simulated annealing. *Science, 220*, 671–680.

Kononenko, I., & Kukar, M. (2007). *Machine learning and data mining*. Woodhead Publishing.

Kuo, R., Amornnikun, P., & Nguyen, T. (2020). Metaheuristic-based possibilistic multivariate fuzzy weighted c-means algorithms for market segmentation. *Applied Soft Computing Journal, 96*.

Lee, C., Huang, K., Shen, Y., & Lee, Y. (2020). Improved weighted k-nearest neighbor based on pso for wind power system state recognition. *Energies*, *13*.

Li, C., Xia, S., Fu, J., Chen, Z., & Wang, B. (2021). An improved genetic algorithm based on k-means. In *2021 International symposium on electrical, electronics and information engineering* (pp. 367–372).

Li, K., Lai, G., & Yao, X. (2023). Interactive evolutionary multi-objective optimization via learning-to-rank. *IEEE Transactions on Evolutionary Computation, 1*–11.

Li, K., Wang, H., Wang, W., Wang, F., & Cui, Z. (2021). Improving artificial bee colony algorithm using modified nearest neighbor sequence. *Journal Of King Saud University Computer And Information Sciences*.

Li, Y., Wang, S., & Yang, B. (2020). An improved differential evolution algorithm with dual mutation strategies collaboration. *Expert Systems with Applications, 153*, 113451.

Li, X., Wu, S., Li, X., Yuan, H., & Zhao, D. (2020). Particle swarm optimization-support vector machine model for machinery fault diagnoses in high-voltage circuit breakers. *Chinese Journal Of Mechanical Engineering (English Edition), 33*.

Li, W., Zhou, Q., & Ren, J., Samantha, S. (2019). Data mining optimization model for financial management information system based on improved genetic algorithm. *Information Systems And E-Business Management,* 1–19.

Martinez, A., Del Ser, J., Villar-Rodriguez, E., Osaba, E., Poyatos, J., Tabik, S., Molina, D., & Herrera, F. (2021). Lights and shadows in evolutionary deep learning: Taxonomy, critical methodological analysis, cases of study, learned lessons, recommendations and challenges. *Information Fusion, 67*, 161–194.

Medress, M., Cooper, F., Forgie, J., Green, C., Klatt, D., O'Malley, M., Neuburg, E., Newell, A., Reddy, D., Ritea, B., Shoup-Hummel, J., Walker, D., & Woods, W. (1977). Speech understanding systems: Report of a steering committee. *Artificial Intelligence, 9*, 307–316.

Mehta, V., Bawa, S., & Singh, J. (2020). Analytical review of clustering techniques and proximity measures. *Artificial Intelligence Review*, 1–29.

Miettinen, K. (1998). *Nonlinear multiobjective optimization*. Springer.

Mitchell, M. (1996). *An introduction to genetic algorithms*. Massachusetts Institute of Technology Press.

Mittal, R., Malik, V., Singh, V., Singh, J., & Kaur, A. (2020). Integrating genetic algorithm with random forest for improving the classification performance of web log data. In *2020 Sixth international conference on parallel, distributed and grid computing (PDGC)* (pp. 177–181).

Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers and Operations Research, 24*, 1097–1100.

Moldovan, D. (2020). Cervical cancer diagnosis using a chicken swarm optimization based machine learning method. In *Proceeding of 2020 international conference on e-health and bioengineering* (pp. 1–4).

Moldovan, D., Anghel, I., Cioara, T., & Salomie, I. (2020). Machine learning in manufacturing: Processes classification using support vector machine and horse optimization algorithm. In *2020 19th RoEduNet conference: networking in education and research (RoEduNet)* (pp. 1–6).

Moldovan, D., Anghel, I., Cioara, T., Salomie, I., Chifu, V., & Pop, C. (2019). Kangaroo mob heuristic for optimizing features selection in learning the daily living activities of people with Alzheimer's. In *2019 22nd international conference on control systems and computer science (CSCS)* (pp. 236–243).

Monnot, J. (2016). Approximation results toward nearest neighbor heuristic. *Yugoslav Journal Of Operations Research, 12*

Montemayor, J., & Crisostomo, R. (2019). Feature selection in classification using binary max-min ant system with differential evolution. In *2019 IEEE congress on evolutionary computation (CEC)* (pp. 2559–2566).

Moodi, M., Ghazvini, M., & Moodi, H. (2021). A hybrid intelligent approach to detect android botnet using smart self-adaptive learning-based pso-svm. *Knowledge-Based Systems, 222*.

Moodi, M., Ghazvini, M., Moodi, H., & Ghavami, B. (2020). A smart adaptive particle swarm optimization-support vector machine: Android botnet detection application. *Journal of Supercomputing, 76*, 9854–9881.

Morrison, D., Jacobson, S., Sauppe, J., & Sewell, E. (2016). Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization, 19*, 79–102.

Nagra, A., Han, F., Ling, Q., Abubaker, M., Ahmad, F., Mehta, S., & Apasiba, A. (2020). Hybrid self-inertia weight adaptive particle swarm optimisation with local search using c4.5 decision tree classifier for feature selection problems. *Connection Science, 32*, 16–36.

Nash, J. (2000). The (dantzig) simplex method for linear programming. *Computing in Science and Engineering, 2*, 29–31.

Neapolitan, R. (2003). *Learning Bayesian networks*. Pearson Prentice Hall.

Nemmich, M., Debbat, F., & Slimane, M. (2019). A data clustering approach using bees algorithm with a memory scheme. In *Lecture notes in networks and systems* (Vol. 50, pp. 261–270).

Nguyen, T., & Kuo, R. (2019). Automatic fuzzy clustering using non-dominated sorting particle swarm optimization algorithm for categorical data. *IEEE Access, 7*, 99721–99734.

Niknam, M., & Thulasiraman, P. (2020). Lpr: A bio-inspired intelligent learning path recommendation system based on meaningful learning theory. *Education and Information Technologies, 25*, 3797–3819. https://doi.org/10.1007/s10639-020-10133-3

Nocedal, J., & Wright, S. (1999). *Numerical Optimization*. Springer.

Ozger, Z., Bolat, B., & Diri, B. (2019). A probabilistic multi-objective artificial bee colony algorithm for gene selection. *JUCS Journal Of Universal Computer Science, 25*, 418–443.

Pacifico, L., & Ludermir, T. (2021). An evaluation of k-means as a local search operator in hybrid memetic group search optimization for data clustering. *Natural Computing, 20*, 611–636.

Page, M., McKenzie, J., Bossuyt, P., Boutron, I., Hoffmann, T., Mulrow, C., Shamseer, L., Tetzlaff, J., Akl, E., & Brennan, S. (2021). Others: The prisma 2020 statement: An updated guideline for reporting systematic reviews. *Systematic Reviews, 10*, 89.

Pan, Z., Wang, Y., & Pan, Y. (2020). A new locally adaptive k-nearest neighbor algorithm based on discrimination class. *Knowledge-Based Systems, 204*, 106185.

Pham, D., Tran, B., Nguyen, S., & Alahakoon, D. (2020). Swarm-based machine learning algorithm for building interpretable classifiers. *IEEE Access, 8*, 228136–228150.

Pourpanah, F., Wang, R., & Lim, C. (2023). Others: A review of artificial fish swarm algorithms: Recent advances and applications. *Artificial Intelligence Review, 56*, 1867–1903.

Pravesjit, S., Longpradit, P., Kantawong, K., Pengchata, R., & Oul, N. (2021). A hybrid pso with rao algorithm for classification of wisconsin breast cancer dataset. In: *2021 2nd international conference on big data analytics and practices (IBDAP)* (pp. 68–71).

Qaddoura, R., Faris, H., & Aljarah, I. (2021). An efficient evolutionary algorithm with a nearest neighbor search technique for clustering analysis. *Journal of Ambient Intelligence and Humanized Computing, 12*, 8387–8412.

Rao, S. (2009). *Engineering optimization: Theory and practice*. Wiley.

Ratanavilisagul, C. (2020). Modification fitness function of particle swarm optimization to improve the cluster centroid. *IOP Conference Series: Materials Science And Engineering, 965*, 012038.

Rokach, L., & Maimon, O. (2005). Clustering methods. In Maimon, O. & Rokach, L. (Eds.), *Clustering methods data mining and knowledge discovery handbook*.

Rokach, L., & Maimon, O. (2008). *Data mining with decision trees: Theory and applications*. World Scientific.

Sarala, S., Chitraivel, M., & Raj, R. (2021). Decision tree based salp swarm optimization for multi medical data classification with feature reduction technique. *Brazilian Archives of Biology and Technology, 64*.

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge University Press.

Shami, T., El-Saleh, A., Alswaitti, M., Al-Tashi, Q., Summakieh, M., & Mirjalili, S. (2022). Particle swarm optimization: A comprehensive survey. *IEEE Access, 10*, 10031–10061.

Shih, Y., & Ting, C. (2019). Evolutionary optimization on k-nearest neighbors classifier for imbalanced datasets. In *2019 IEEE congress on evolutionary computation (CEC)* (pp. 3348–3355).

Singh, H., & Kumar, Y. (2020). A neighborhood search based cat swarm optimization algorithm for clustering problems. *Evolutionary Intelligence, 13*, 593–609.

Singh, T. (2021). A novel data clustering approach based on whale optimization algorithm. *Expert Systems, 38*.

Sivanandam, S., & Deepa, S. (2008). *Introduction to genetic algorithms*. Springer.

Sohrabi, M., & Azgomi, H. (2020). A survey on the combined use of optimization methods and game theory. *Archives of Computational Methods in Engineering, 27*, 59–80.

Storn, R., & Price, K. (1997). Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. *Journal Of Global Optimization, 11*, 341–359.

Sulaiman, M., Mustaffa, Z., Zakaria, N., & Saari, M. (2023). Using the evolutionary mating algorithm for optimizing deep learning parameters for battery state of charge estimation of electric vehicle. *Energy, 279*, 128094.

Suyanto, S., Wibowo, A., Faraby, S., Saadah, S., & Rismala, R. (2021). Evolutionary rao algorithm. *Journal of Computational Science, 53*, 101368.

Suyyagh, A., Tong, J., & Zilic, Z. (2016). Performance evaluation of meta-heuristic in energy aware real-time scheduling problems. *Jordanian Journal of Computers and Information Technology, 2*, 70–87.

Telikani, A., Tahmassebi, A., Banzhaf, W., & Gandomi, A. (2021). Evolutionary machine learning: A survey. *ACM Computing Survey*, *54*.

Thalamala, R., Reddy, A., & Janet, B. (2020). A novel bio-inspired algorithm based on social spiders for improving performance and efficiency of data clustering. *Journal of Intelligent Systems, 29*, 311–326.

Tharwat, A., & Gabel, T. (2020). Parameters optimization of support vector machines for imbalanced data using social ski driver algorithm. *Neural Computing Application, 32*, 6925–6938.

Tharwat, A., & Hassanien, A. (2018). Chaotic antlion algorithm for parameter optimization of support vector machine. *Applied Intelligence, 48*, 670–686.

Tharwat, A., & Hassanien, A. (2019). Optimizing support vector machine parameters using bat optimization algorithm. In *Studies in computational intelligence* (Vol. 801, pp. 351–374).

Theodoridis, S. (2015). *Machine learning: A Bayesian and optimization perspective*. Academic Press.

Velliangiri, S., Alagumuthukrishnan, S., & Thankumar joseph, S. (2019). A review of dimensionality reduction techniques for efficient computation. *Procedia Computer Science, 165*, 104–111.

Voudouris, C., & Tsang, E. (1996) Partial constraint satisfaction problems and guided local search. In *Proceedings of second international conference on practical application of constraint technology (PACT'96)* (pp. 337–356).

Wang, H., Wang, W., Xiao, S., Cui, Z., Xu, M., & Zhou, X. (2020). Improving artificial bee colony algorithm using a new neighborhood selection mechanism. *Information Sciences, 527*, 227–240.

Wang, S., Li, Y., & Yang, H. (2019). Self-adaptive mutation differential evolution algorithm based on particle swarm optimization. *Applied Soft Computing, 81*, 105496.

Wang, X., Zhang, F., & Yao, M. (2023). A many-objective evolutionary algorithm with estimating the convexity-concavity of pareto fronts and clustering. *Information Sciences, 644*, 119289.

Willis, M., Zhang, L., Liu, H., Xie, H., & Mistry, K. (2020). Object recognition using enhanced particle swarm optimization. In *2020 International conference on machine learning and cybernetics (ICMLC)* (pp. 241–246).

Wolpert, D., & Macready, W. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 1*, 67–82.

Wu, X. (2019). Research on the influence of crossover probability and mutation probability in ga-svm model. In *2019 IEEE 9th international conference on electronics information and emergency communication (ICEIEC)* (pp. 1–7).

Wuerges, A., & Borba, J. (2010). Redes neurais, lógica nebulosa e algoritmos genéticos: Aplicações e possibilidades em finanças e contabilidade. *Revista De Gestão Da Tecnologia E Sistemas De Informação, 7*, 163–182.

Yang, C., Tu, X., & Chen, J. (2007). Algorithm of marriage in honey bees optimization based on the wolf pack search. In *The 2007 International conference on intelligent pervasive computing (IPC 2007)* (pp. 462–467).

Yang, X. (2010). *Engineering optimization: An introduction with metaheuristic applications*. W iley.

Yang, X. (2010). Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-inspired Computation, 2*, 78–84.

Yang, X., He, X., & Fan, Q. (2020). Chapter 7 -mathematical framework for algorithm analysis. In *Nature-inspired computation and swarm intelligence* (pp. 89–108).

Yu, J., Wang, M., J. H., Y., & Arefzadeh, S. (2023). A new approach for task managing in the fog-based medical cyber-physical systems using a hybrid algorithm. *Circuit World*, *49*, 294–304.

Yu, X., Zhang, Z., & Song, M. (2021). Short-term wind speed prediction of wind farms based on particle swarm optimization support vector machine. In *Proceeding Of IOP conference series: Earth and environmental science* (Vol. 804 , No. 3).

Zgurovsky, M., Sineglazov, V., & Chumachenko, E. (2021). *Artificial intelligence systems based on hybrid neural networks - theory and applications*. Switzerland: Springer.

Zhang, X., Lin, Q., Mao, W., Liu, S., Dou, Z., & Liu, G. (2021). Hybrid particle swarm and grey wolf optimizer and its application to clustering optimization. *Applied Soft Computing, 101*, 107061.

Zhao, L., Gai, M., & Jia, Y. (2019). Classification of multiple power quality disturbances based on pso-svm of hybrid kernel function. *Journal of Information Hiding and Multimedia Signal Processing, 10*, 138–146.

Zhou, M., Lin, F., Hu, Q., Tang, Z., & Jin, C. (2020). Ai-enabled diagnosis of spontaneous rupture of ovarian endometriomas: A pso enhanced random forest approach. *IEEE Access, 8*, 132253–132264.

Zhou, Y., Wu, H., Luo, Q., & Abdel-Baset, M. (2019). Automatic data clustering using nature-inspired symbiotic organism search algorithm. *Knowledge-Based Systems, 163*, 546–557.

Zhu, S. (2019). Research on data mining of education technical ability training for physical education students based on a priori algorithm. *Cluster Computing, 22*, 14811–14818.

Authors and Affiliations

Beatriz Flamia Azevedo[1,2,3] · **Ana Maria A. C. Rocha[3]** · **Ana I. Pereira[1,2,3]**

✉ Beatriz Flamia Azevedo
beatrizflamia@ipb.pt

Ana Maria A. C. Rocha
arocha@dps.uminho.pt

Ana I. Pereira
apereira@ipb.pt

[1] Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, 5300-252 Bragança, Portugal

[2] Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC), Instituto Politécnico de Bragança, 5300-253 Bragança, Portugal

[3] ALGORITMI Research Centre/LASI, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal