The Case for Hybrid Multi-Objective Optimisation in High-Stakes Machine Learning Applications

Alex A. Freitas
University of Kent, UK
School of Computing, University of Kent
Canterbury, CT2 7FS, United Kingdom

ABSTRACT

Most classification (supervised learning) algorithms optimise a single objective, typically the predictive performance of the learned classification model. However, in high-stake classification applications, involving e.g. decisions about whether or not an individual should undergo a medical surgery, be granted a loan or be hired for a job, often there is a need to optimise multiple objectives, such as the predictive performance, interpretability or fairness of the learned model. In this context, this position paper discusses the pros and cons of two different multi-objective optimisation approaches (the Pareto and the lexicographic approaches), and proposes a conceptual framework for hybrid multi-objective optimisation, combining those two approaches.

Keywords

Classification, multi-objective optimisation, Pareto dominance, lexicographic optimisation.

1. INTRODUCTION

Classification algorithms, a major type of supervised machine learning algorithms [39], [64] are currently ubiquitously applied in a wide range of application domains; including domains that involve high-stakes decisions about people, e.g. predicting who should be granted a loan, hired for a job, undergo a surgery, etc. In such applications, it is often desirable that a classification algorithm should optimise not only predictive accuracy but also several other quality criteria of the learned model, such as its interpretability, fairness, etc. Optimising these criteria separately, one at a time, is in general not a good option, since there are usually strong trade-offs between different types of criteria – for instance, the trade-offs between accuracy and interpretability [12], [42], [62], [52] between accuracy and fairness [63], [1], [49], [59], between accuracy and inference time [65], [31], and between accuracy and privacy [8], [50]. Hence, there is a clear need for multi-objective optimisation methods that optimise multiple criteria (objectives) at the same time.

Furthermore, for each of these broad types of criteria (e.g. accuracy, interpretability, fairness), there are usually multiple specific measures of the quality of a predictive model measuring different aspects of that criterion – discussed e.g. in [25], [30], [37] for predictive accuracy measures; [38], [10], [60] for fairness measures; and [6], [41] for interpretability measures. Each of such specific measures of a model's quality can also be considered as a separate objective to optimised, leading again to the need for multi-objective optimisation methods to obtain more robust results. For example, there is no predictive accuracy measure which is universally superior to all other measures, with different accuracy measures having different pros and cons [21], [23], [44]; and so, in practice it makes sense to try to optimise more than one accuracy measures, to perform a more robust evaluation of predictive accuracy. There are also trade-offs between different measures of interpretability [48], [40] and different measures of fairness [2], [29], [9].

The need for multi-objective optimisation also arises naturally in several types of machine learning (sub)-areas. For example, multi-task learning problems in general can be naturally cast as multi-objective optimisation problems [53], where predictive accuracy in each task is an objective to be optimised. In addition, in the area of multi-label classification, which is a specific type of multi-task learning, it is standard procedure to report the results of multiple measures of predictive accuracy, since no measure captures all the nuances of multi-label classification performance [58], [45], [4]. Optimising multiple multi-label predictive accuracy measures can be naturally cast as a multi-objective optimisation problem. As another example, in federated learning [33], since the data and model computation have to be distributed across many local processors, including low-speed, low-memory local devices, objectives to be optimised include predictive accuracy, model complexity, communication costs and memory requirements on local devices [66].

Yet another machine learning area with a strong and natural need for multi-objective optimisation is Automated Machine Learning (Auto-ML), which essentially consists of using an optimisation method to search for the best learning algorithm (or pipeline) and its best hyper-parameter settings for an input dataset [3], [26], [67]; where, in the literature, "best" usually means "most accurate" based on a given objective function. However, given the very large and heterogenous search space of Auto-ML systems, there is a clear motivation to optimise not only predictive accuracy but also the computational resources (e.g. time) to learn each classifier, leading to 'resource-aware multi-objective optimisation' [65]. This is particularly relevant in the area of neural architecture search, a sub-area of Auto-ML where the search space includes (deep) neural network architectures – whose training usually requires a large amount of time and memory [24], [66]. In this scenario, multi-objective optimisation has been used to simultaneously optimise predictive accuracy and other objectives such as a network's inference time [28], [15], [16], a network's number of parameters [16], [15] or number of floating point operations / multiply-add operations [15], [36], [16], or memory usage on mobile phones [15].

Despite this clear need for multi-objective evaluation of predictive models in a wide range of classification problems, the vast majority of the literature still focus on the traditional framework of single-objective optimisation, focusing mainly on predictive accuracy – and often a single measure of predictive accuracy.

When multiple objectives are optimised in supervised learning, this is usually implemented by converting the original multi-objective problem into a single-objective one by using a linear combination (weighted sum) of the original objectives of the form: $w_1 \times Obj_1 + \ldots + w_m \times Obj_m$, where w_i, $i = 1,\ldots,m$, denotes the weight assigned to objective Obj_i, and m is the

number of objectives to be optimised. This approach has the advantage of conceptual simplicity, but it also has clear disadvantages: it requires the specification of *ad-hoc* weights to each objective, and each run of the optimisation algorithm considers only one possible trade-off among the objectives. In practice, to consider multiple trade-offs, users could run the algorithm many times by varying the objectives' weights across the runs, but this is inefficient (very time-consuming) and ineffective [13], [11], [19], since each run of the algorithm ignores valuable information about the qualities of candidate solutions evaluated in previous runs of the algorithm.

This article focuses instead on two genuinely multi-objective optimisation approaches, namely the Pareto and the lexicographic approaches [13], [18]. Both approaches have the advantage of exploring multiple trade-offs between the different objectives in a single run of the optimisation algorithm, avoiding the need for mixing different objectives into a linear combination of weighted objectives. In essence, the Pareto approach finds a set of 'non-dominated solutions' (the Pareto front) where, for each solution s in the Pareto Front, there is no other solution that performs better than s for at least one objective and performs at least as well as s for all other objectives; whilst the lexicographic approach optimises the multiple objectives in decreasing order of their priorities. These approaches will be reviewed in Section 2.

In the literature on multi-objective optimisation for machine learning, the Pareto approach is in general much more popular than the lexicographic approach. Actually, the Pareto approach is often presented as the only good approach to avoid the disadvantages of the weighted sum approach, and the Pareto approach's limitations are often ignored or downplayed; whilst the lexicographic approach is often ignored. As evidence for this, several surveys of multi-objective optimisation do not even mention the lexicographic multi-objective optimisation approach [56], [57], [34], [35], [43], [54].

In this context, this position article has two contributions. The first one is to show that the Pareto and the lexicographic approaches have to a large extent complementary pros and cons, i.e., none of them is inherently better than the other; and in real-world applications, their use should be determined mainly by the needs and interests of users and the requirements of the target application domain. The second contribution is to propose a new conceptual, hybrid multi-objective optimisation framework designed for synergistically combining the best aspects of the Pareto and lexicographic approaches, in order to offer users an effective and flexible approach for multiple objective optimisation – particularly in the context of high-stakes real-world machine learning applications, where there is a strong need for optimising multiple objectives, as discussed earlier.

The remainder of this article is organised as follows. Section 2 briefly reviews background on the Pareto and lexicographic approaches, to make this article more self-contained. Section 3 discusses the pros and cons of these two approaches. Section 4 described the proposed conceptual, hybrid framework for multi-objective optimisation. Section 5 reports the conclusions.

2. BACKGROUND

The Pareto approach is based on the concept of Pareto dominance between candidate solutions (classifiers, in this article). When comparing two classifiers, a classifier C_1 dominates another classifier C_2 if and only if: C_1 is better than C_2 with respect to at least one objective, *and* C_1 is not worse than C_2 with respect to all the objectives [13], [17], [18]. More formally, let $f_i(C_j)$ denote the value of the i-th objective for classifier C_j. Assuming, without loss of generality, that all m objectives are to be maximised, a classifier C_1 dominates another classifier C_2 if and only if: $\exists i$ such that $f_i(C_1) > f_i(C_2)$ and $\forall i$, i=1,…,m, $f_i(C_1) \geq f_i(C_2)$; where m is the number of objectives being optimised. A classifier is said to be non-dominated if it is not dominated by any other classifier.

The concept of Pareto dominance is illustrated in Figure 1, using as an example a hypothetical case where there are two objectives to be maximised, namely the predictive accuracy of a classifier and the fairness of its predictions. In Figure 1, classifier B is clearly dominated by classifier A, which has better accuracy and better fairness. Likewise, classifier D is dominated by classifier C. Classifier E is also dominated by classifier C, because, although classifiers C and E have the same accuracy, C has better fairness, which satisfies the aforementioned definition of Pareto dominance. In addition, classifier G is dominated by classifiers C, E, F. Finally, classifiers A, C, F are non-dominated, and they form the Pareto front in the context of the 7 classifiers in this simple example.

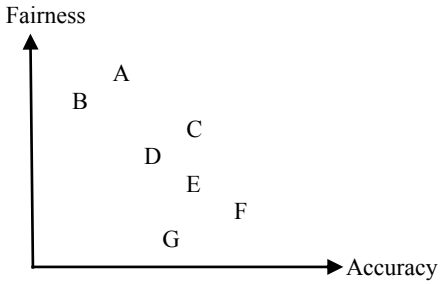

Figure 1: Examples of Pareto dominance

In the Pareto approach, in general the optimiser aims at finding the set of all non-dominated classifiers. However, the only way to guarantee that all non-dominated solutions are found would be to perform an exhaustive search evaluating all candidate solutions in the search space, which is not feasible in general. Hence, in practice Pareto-based optimisers return the best estimate of the set of non-dominated solutions that they were able to find within their computational budget. In most works in this area, it is simply assumed that all the non-dominated solutions found by the optimiser can be returned to the user and that the user would then presumably choose one of those solutions to be deployed in the real-world, based on the user's preferred trade-off among the multiple objectives [13], [27] (the pros and cons of leaving such choice to the user are discussed later). In some works, however, the optimiser returns only a subset of the found non-dominated solutions to simplify the user's analysis of those solutions, as discussed later.

The lexicographic approach requires the user to define a priority ordering for the objectives, and then the objectives are optimised in decreasing order of priority [18], [68], [20], [5]. That is, in order to select the best out of two classifiers, they are first compared with respect to the first (highest-priority) objective. If one classifier is better than the other regarding that objective, the former is declared the winner. Otherwise (i.e. there is a "tie" in the objective values of the two classifiers), the two classifiers are

compared with respect to the second objective. Again, if one classifier is better than another regarding that objective, the former is declared the winner, and so on, until a winner is chosen. When comparing classifiers, the choice of a winner depends on how "a tie" is defined for two values of an objective. In the simpler case of objectives with discrete values, a tie can be defined as two classifiers having exactly the same discrete value for an objective. However, in machine learning it is more common to have real-valued objectives, and in this case a tie is usually defined as a difference of objective values that is smaller than or equal to a small ε (a "tolerance threshold"), so that a classifier is "better than" another regarding an objective only if the difference in their objective values is greater than ε. Finally, if two classifiers are tied regarding all objectives based on the tolerance threshold, the best classifier can be chosen as the one with the best value of the first objective, ignoring the tolerance threshold.

To clarify the use of the lexicographic approach, let us consider a hypothetical case where again there are two objectives to be maximised, namely the predictive accuracy of a classifier (Acc) and the fairness of its predictions (Fairn), both objectives taking a value in the range [0…1] for each classifier. Assume that the user specified that maximizing Acc has priority over maximizing Fairn, and the tolerance threshold is ε = 0.01.

Consider now two classifiers: C_1, with Acc = 0.7 and Fairn = 0.8; and C_2, with Acc = 0.9 and Fairn = 0.6. When comparing classifiers C_1 and C_2 based on the lexicographic optimization approach, C_2 is declared the better classifier because it has substantially better Acc, i.e., the difference between C_2's Acc and C_1's Acc, 0.2 (0.9 – 0.7), is greater than ε (0.01). In this case, the fact that C_1 has substantially better fairness does not affect the result of the lexicographic comparison, because C_2 won over C_1 in the higher-priority objective of accuracy, so there is no need to consider the lower-priority objective of fairness.

Extending the previous example, consider now a classifier C_3, with Acc = 0.69 and Fairn = 0.85, and the classifier C_1 of the previous example (with Acc = 0.7 and Fairn = 0.8). Now, when comparing classifiers C_1 and C_3 based on the lexicographic optimization approach, they are "tied" in the higher-priority Acc objective, i.e. there is no substantial difference in their Acc values, since their Acc difference of 0.01 is not greater than the tolerance threshold ε (0.01). Hence, C_1 and C_3 need to be compared in terms of the lower-priority objective of fairness. In this case, C_3 has a substantially better Fairn value than C_1, with a difference of 0.05 (0.85 – 0.8), which is greater than the tolerance threshold ε (0.01). Therefore, C_3 is declared the winner of the lexicographic comparison; meaning that, in this case, it is acceptable to incur a small, non-substantial (1%) loss of accuracy in order to achieve a substantial gain of fairness, based on the user-defined priority order of the objectives and tolerance threshold.

Several examples of the use of the Pareto and lexicographic approaches in the classification task will be given in Section 4, where a hybrid Pareto/lexicographic multi-objective optimization framework is proposed.

3. PROS AND CONS OF THE PARETO AND LEXICOGRAPHIC APPROACHES

This section discusses the pros and cons of these two approaches in the context of two main issues: (a) how each approach copes with users' preferences about different objectives (Section 3.1); and (b) how users cope with the solution(s) returned by the multi-objective optimizer (Section 3.2).

3.1 Coping with Users' Preferences About Different Objectives

First, since the Pareto approach is agnostic regarding the relative importance of the objectives, it is a natural choice in scenarios where the user does not have any preference about the objectives. This partly explains the popularity of the Pareto approach in the academic literature. In many papers on multi-objective machine learning, the authors are data analysts with expertise on machine learning, rather than users with expertise on the data and its application domain, and the learned models are not used to make decisions in the real-world. In this context of academic research, it is intuitively appealing to data analysts to use the Pareto approach, which avoids the need to decide how to prioritise some objective(s) over others in the real-world.

In many real-world applications, however, users may naturally want to prioritise the optimisation of some objective(s) over others. For example, intuitively most users would prioritise the optimisation of a model's predictive accuracy over other objectives, like a model's interpretability or fairness; whilst some users might prioritise, e.g., fairness or privacy even over accuracy, if there is a legal requirement for fairness or privacy. In scenarios where users can easily specify a clear priority ordering for multiple objectives, the lexicographic approach is intuitively more natural, since it allows the optimisation algorithm to take the very important user preferences into account, whilst those preferences would be ignored by the Pareto approach [5]. It should also be noted that, in practice, it is usually much easier for users to specify a (qualitative) priority order for objectives than specifying the precise numerical (quantitative) weights for all objectives as required in the weighted-sum approach. For example, it is natural for a user to say that predictive accuracy has priority over model size; but it would be much harder for a user to justify why the weights for accuracy and model size should be e.g. 0.8 and 0.2, or 0.67 and 0.33, or whatever other weights.

In addition, a point that is usually ignored in the Pareto optimisation literature is that often the user will be interested in just a region of the Pareto front [19], [61], [47], and in such cases searching for the entire Pareto front would involve a waste of computational resources. For example, in the common scenario where maximising predictive accuracy has priority over minimising model size, a model with the smallest possible size and very low accuracy might be selected and remain in the Pareto front (to be compared against other models for updating the Pareto front) for many iterations of the optimiser, despite being clearly an unacceptable solution to users. In general, such a model would not be selected by the lexicographic approach, due to its very low accuracy (as the higher-priority objective).

On the other hand, an argument commonly used against the lexicographic approach is that, unlike the Pareto approach, the lexicographic approach has the disadvantage of requiring the specification of ad-hoc tolerance thresholds. At first glance, one could argue that, *in theory*, such tolerance thresholds are about as much ad-hoc as the numerical weights for each objective in the baseline weighted-sum approach. Actually, in the lexicographic approach, broadly speaking, other things being equal, an objective's importance is inversely proportional to its tolerance threshold value – since the smaller the tolerance threshold for an

objective, the fewer the "ties" between two values of that objective (for two solutions), meaning that the objective will be used more often to choose the winner solution when comparing two candidate solutions.

However, as the old saying goes: "in theory there is no difference between theory and practice, but in practice there is". In practice, the tolerance thresholds of the lexicographic approach are less problematic than the numerical weights of the weighted-sum approach, as follows.

First, there is in principle no need for any tolerance threshold when an objective to be optimised takes discrete values (like e.g. the objective of minimising the depth or size of a decision tree), since in this case there is a natural "tie" between two solutions when they have exactly the same discrete value of that objective. However, as mentioned earlier real-valued objectives are more common in machine learning; and some tolerance threshold is required when comparing two classifiers regarding a real-valued objective, for two reasons: in practice a difference very close to zero tends to be irrelevant, and strict equality is not a good operator to use when comparing two real-valued numbers in a computer (with finite arithmetic).

Note, however, that although the tolerance thresholds of the lexicographic approach have the effect of performing some fine-tuning of the relative importance of the different objectives, in practice the relative importance of an objective in this approach is still by far primarily determined by its position in the ordered priority list. In theory we could use a tolerance threshold to radically change an objective's importance, e.g. if we set the tolerance threshold of the highest priority objective to infinite, then there would always be a tie in that first objective, which would eliminate that objective's importance. In practice, however, no one sets tolerance threshold to infinite or even large values, tolerance thresholds are in general simply set to small values, say from about 1% to 5% of the range of values for an objective. With such "reasonably small values", tolerance thresholds have much less influence on the relative importance of different objectives than the user-specified priority order of objectives (which is an effective form of incorporating the user's preferences into the optimiser).

Another point is that, as long as different objectives have been normalised to the same range of values, in many cases it seems reasonable to specify a single value of a tolerance threshold for all (real-valued) objectives, rather than different values for different objectives. This substantially reduces the number of "ad-hoc" parameters.

In summary, arguably the need for specifying tolerance thresholds still counts as a disadvantage of the lexicographic approach, by comparison with the Pareto approach (which does not use such thresholds), but this disadvantage is in general substantially smaller than the disadvantage of having to specify ad-hoc numerical weights for the objectives in the weighted-sum approach. In addition, the use of tolerance thresholds is usually a price worth paying for the benefit of directly specifying the user's relative preferences for the multiple objectives to be optimised, in cases where the user has clear preferences (which would be ignored in the standard Pareto approach).

Note also that, although the Pareto approach does not *explicitly* require any parameter to cope with the users' preferences about different objectives, in practice, at the algorithm level, in order to search for the best Pareto front, a Pareto-based optimiser usually has some *implicit* parameters associated specifically with the Pareto optimisation process. For example, the NSGA-II algorithm [14], probably the most popular Pareto-based optimiser, uses a "crowding" procedure that encourages diversity in the non-dominated solutions in the Pareto front maintained by the algorithm along its search. It is claimed in [14] that this procedure does not require any user-specified parameter, but this procedure involves at least the choices of a distance function and a normalisation procedure for distance computation, which in practice can be considered implicitly user-specified parameters.

3.2 Coping with the Solution(s) Returned by the Multi-Objective Optimiser

In the Pareto approach the optimiser returns a set of non-dominated solutions, since the Pareto dominance concept is completely agnostic with respect to the relative importance of the different objectives, and so there is no clearly "best" solution among all the non-dominated solutions. As mentioned earlier, the standard approach for coping with a large number of non-dominated solutions returned to the user is to simply assume that it is up to the user to choose a single best among all non-dominated solutions using their own subjective preference [13], [27]. This approach is usually acceptable in academic research where the solutions returned by the optimiser will not be deployed in the real world.

However, in real-world applications, this approach can be regarded as a double-edged sword, considering that in many applications ultimately a single solution needs to be chosen for practical reasons. On one hand, returning many non-dominated solutions provides more flexibility to users, giving them the chance of using their subjective evaluation of the pros and cons of different solutions (i.e. the extent to which different measures were optimised) to choose the best solution. Importantly, since the user makes this choice by considering a set of "high-quality" non-dominated solutions *a posteriori* (after the optimiser returned its results), this is a much more well-informed choice than the much less well-informed choice of ad-hoc weights for each objective *a priori* (before running the optimiser) in the weighted-sum approach [13], [18].

On the other hand, users may find it difficult to subjectively choose among a large (often very large) set of non-dominated solutions. There are automated methods for choosing a subset of "the best" non-dominated solutions [46], [55], [65], [32], so that the user could focus their attention on a relatively small set of most promising solutions. However, there is no guarantee that such methods will choose the solution that would be really the best solution for the user in practice, since such methods tend to ignore users' preferences.

By contrast, in the lexicographic approach the optimiser returns a single optimised solution, representing the best trade-off among the objectives found by the lexicographic optimiser, which took into account the user's priority ordering of objectives.

Table 1 summarises the above discussion on the main differences between the Pareto and lexicographic approaches. Note that these two approaches have largely complementary pros and cons, i.e. none is inherently superior to the other.

Table 1: Summary of the main differences between the Pareto and lexicographic approaches for multi-objective optimization, with their complementary pros and cons

Issue 1: How the optimiser copes with users' preferences about different objectives	
Pareto approach	Lexicographic approach
Agonistic about users' preferences for objectives; optimiser searches for all non-dominated solutions	Optimises objectives in decreasing order of priority, which is specified by the user
Pro: no parameter required for representing users' preferences	**Pros:** Incorporates users' preferences for objectives as background knowledge; optimiser focuses on solution space region more interesting for users
Con: optimiser can waste time finding solutions in Pareto front regions not relevant for users	**Con:** Requires tolerance-threshold parameters for real-valued objectives (not necessarily for discrete objectives)
Issue 2: How the user copes with the solution(s) returned by the optimiser	
Pareto approach	Lexicographic approach
Optimiser returns a set of non-dominated solutions; user chooses preferred non-dominated solution *a posteriori*	Optimiser returns a single solution to the user
Pro: provides users with flexibility for choosing their preferred solution	**Pros:** the returned solution was chosen based on the users' priorities for different objectives; user does not need to spend time or to make a difficult decision for selecting a solution among many non-dominated solutions
Con: users may find it difficult to select a solution from a (often very large) set of non-dominated solutions	**Con:** users cannot evaluate the different trade-offs among objectives in multiple non-dominated solutions

4. A FRAMEWORK FOR HYBRID PARETO AND LEXICOGRAPHIC MULTI-OBJECTIVE OPTIMISATION

In the literature on multi-objective optimisation (MOO) in machine learning, normally authors simply use either the Pareto or the lexicographic approach (much more often the former), without considering the possibility of combining these two approaches to improve the effectiveness of the MOO optimiser. To address this gap, this section proposes a framework for creating hybrid MOO optimisers, to try to synergistically combine 'the best of both worlds' into a more effective MOO optimiser.

In the proposed framework, the multiple objectives to be optimised are divided into groups. The framework is designed to be flexible about how the objectives are divided into groups. This is a task that should be performed by the user, based on their expertise and subjective preferences regarding which objectives should be prioritised over others (using the lexicographic approach) in some group(s) and which objectives should be optimised without specifying their relative priorities (using the Pareto approach) in other group(s).

In the case of real-world applications, in general the user would be the person who would use the predictions of the learned models to make decisions in the real world, and ideally the user would be an expert on the data and its application domain. In purely academic research, without real-world applications and without access to real world users, the role of the user would be simulated by the data analyst, usually the authors of the paper, who typically have expertise on machine learning.

When creating the groups of objectives, there are two types of decisions to be made by the user, about which type of MOO approach should be used. First, at the 'within-group' level, for each group of objectives, the user specifies the type of MOO approach (i.e., the Pareto or the lexicographic approach) to be used to optimise objectives in that group. Different groups can use different types of MOO approaches, but all objectives within a group will be optimised by the same type of MOO approach. Second, at the 'across-groups' level, the user specifies the type of MOO approach to be used for the joint optimisation of all groups of objectives as a whole.

These two types of decisions lead to four possible scenarios, summarised in Table 2. When the Pareto approach is used at the across-groups level (Scenarios 1 and 2), at the within-group level we can have either have a homogenous use of the lexicographic approach, i.e. it is used within all groups of objectives (Scenario 1); or a heterogeneous use of the Pareto and lexicographic approaches, i.e. some group(s) of objectives use one of these approaches whilst other group(s) use the other approach (Scenario 2). Analogously, when the lexicographic approach is used at the across-groups level (Scenarios 3 and 4), at the within-group level we can have either a homogeneous use of the Pareto approach in all groups of objectives (Scenario 3) or a heterogeneous use of the Pareto and lexicographic approaches (Scenario 4). Note that we do not consider the trivial scenarios where one approach (Pareto or lexicographic) is used at the across-groups level and the same approach is used in every group at the within-group level because these scenarios would *not* lead to any hybrid MOO approach.

Table 2: Four scenarios for a hybrid Pareto and lexicographic multi-objective optimization (MOO) approach

MOO scenario	Across-groups MOO approach	Within-group MOO approach(es)
1	Pareto	Homogeneous lexicographic
2		Heterogeneous Par & Lex
3	Lexicographic	Homogeneous Pareto
4		Heterogeneous Par & Lex

In the remainder of this paper, to simplify the discussion of the scenarios shown in Table 2, we will refer to two groups of objectives, each group containing only two objectives (i.e. 4 objectives in total). In practice, 4 objectives might often be

enough to give users a reasonably robust multi-criteria perspective on the performances of different classifiers, the kind of perspective usually missing in the literature. However, if necessary, the ideas proposed in this paper can be naturally extended to more complex scenarios with more than two groups of objectives and/or more than two objectives per group.

Scenario 1: Pareto approach at the across-groups level and homogeneous use of the lexicographic approach at the within-group level

In this scenario, when two classifiers are compared, first, for each group of objectives, a lexicographic optimiser determines the winner classifier using the lexicographic approach. Then, the Pareto approach is used by the optimiser at the across-groups level in order to determine if one of the classifiers dominates the other.

More precisely, in the Pareto optimiser at the across-groups level, a classifier C_1 dominates a classifier C_2 if and only if: (C_1 is lexicographically better than C_2 in *at least one group* of objectives) and (C_1 is lexicographically better than or tied with C_2 in *all groups* of objectives). In other words, within each group of objectives there is a lexicographic comparison between C_1 and C_2 based on the objectives in that group, and a classifier will be a winner at the across-groups level when that classifier is a lexicographical winner within at least one group of objectives and that classifier is not a lexicographical loser in any of the groups of objectives.

Note that in this scenario the Pareto approach is applied to the *qualitative* results of the lexicographic approach applied to each group of objectives, rather than the *numerical values* of the individual objective functions (like in the standard definition of Pareto dominance, in Section 2).

Conceptual Example for Scenarios 1 and 2: Consider a classification task where the class variable indicates whether or not the patient has a specific type of cancer, with 4 objectives to be optimised, divided into 2 groups (2 objectives per group). The first group has two objective functions measuring predictive accuracy: Recall and Precision of the class: 'Cancer=yes'. The user decided that maximizing Recall has higher priority than maximizing Precision, because it is more important reducing the number of false negatives (cancer patients wrongly classified as no-cancer patients) than reducing the number of false positives (no-cancer patients wrongly classified as cancer patients) – since a false negative result is more likely to lead to the death of a patient (due to not treating a cancer patient) than a false positive result. The second group has two objective functions measuring a classifier's interpretability: the degree of violation of monotonicity constraints by the classifier [7], [22], and the classifier's size. The user decided that minimizing the classifier's violation of monotonicity constraints (related to domain knowledge) has higher priority than minimising the classifier's size (a purely syntactic measure of simplicity).

Numerical Example for Scenarios 1 and 2: Consider two classifiers C_1 and C_2, whose values for each of the above 4 objectives are as shown in Table 3. Assume that, for all objectives, the tolerance threshold for the lexicographic approach is 0.01. Regarding the two objectives in group 1, there is no substantial difference between the classifiers C_1 and C_2 regarding the higher-priority recall measure (the difference of their recalls is within the tolerance threshold of 0.01), and classifier C_2 has a substantially higher precision; so C_2 wins the lexicographic comparison in group 1. Regarding the two objectives in group 2, C_2 has a substantially smaller degree of violation of monotonicity constraints, which is the higher-priority objective in group 2, and so C_2 also wins the lexicographic comparison in group 2. Then, comparing C_1 and C_2 across the two groups of objectives using the Pareto approach, based on the qualitative results of the lexicographic comparisons within each group, C_2 is lexicographic better than C_1 in both groups 1 and 2, so C_2 dominates C_1.

Table 3: Example for the use of the hybrid framework in scenarios 1 and 2

Classifier	Objectives in group 1		Objectives in group 2	
	Recall	Precision	Monot-Viol	Size
C_1	0.61	0.50	0.50	0.30
C_2	0.60	0.65	0.45	0.50

Note that in this example of scenario 1, the result of the hybrid MOO approach, i.e. C_2 dominates C_1, is very different from the result that we would obtain if we simply applied the Pareto approach to all 4 objectives in Table 3, in which case neither of C_1 or C_2 would dominate the other, i.e., they would be both non-dominated. This example also illustrates the fact that, broadly speaking, as the number of objectives grows, it becomes harder to find a solution that dominates others, and so there is an increasing tendency to have larger sets of non-dominated solutions, potentially a problem for users that have to select one out of a large number of non-dominated solutions, as mentioned earlier. In this example, the application of the lexicographic approach at each of the two smaller groups of objectives allowed the Pareto-based optimiser at the across-groups levels to conclude that C_2 clearly dominates C_1, since C_2 won the lexicographic comparisons in both group 1 (accuracy-related objectives) and group 2 (interpretability-related objectives). This is arguably an intuitively better result, based on the user's declared preferences in each of the two groups of objectives.

Scenario 2: Pareto approach at the across-groups level and heterogenous use of the Pareto and lexicographic approaches at the within-group level

In this scenario the user has chosen to use the Pareto approach at the across-groups level (like Scenario 1), and has chosen to use the lexicographic approach in some group(s) and the Pareto approach in other group(s) of objectives, at the within-group level. Since our running example (Table 3) has only two groups, we have to consider only two cases in this scenario, as follows.

Case (A): lexicographic approach in group 1 and Pareto approach in group 2: In group 1, classifier C_2 wins the lexicographic comparison as mentioned earlier for scenario 1. In group 2, classifiers C_1 and C_2 are non-dominated (neither dominates the other). Therefore, the Pareto optimiser at the across-groups level considers that C_2 is better than C_1 in group 1 and there is a tie between C_1 and C_2 in group 2, concluding that C_2 dominates C_1.

Case (B): Pareto approach in group 1 and lexicographic approach in group 2: In group 1, classifiers C_1 and C_2 are non-dominated. In group 2, classifier C_2 wins the lexicographic comparison as mentioned earlier for Scenario 1. Therefore, the Pareto optimiser at the across-groups level considers that there is a tie between C_1 and C_2 in group 1 and C_2 is better than C_1 in group 2, concluding again that C_2 dominates C_1.

In the example of Table 3, the Pareto optimiser at the across-groups level obtained the same result in both case (A) and case (B), because the lexicographic comparisons in both group 1 and group 2 consistently return the result of C_2 being better than C_1, and when the lexicographic approach is replaced by the Pareto in one of the two groups, although there is tie (non-dominance) in that group, the lexicographic win of C_2 in the other group is enough to make C_2 win based on the Pareto approach at the across-groups level.

Note, however, that this kind of result pattern does not generalize to all uses of this scenario. For example, suppose the Recall of classifier C_1 in Table 3 was 0.62 (or higher), and all other data in Table 3 remained the same. Then, C_1 would be lexicographically better than C_2 in the group 1 of objectives, and C_2 would be lexicographically better than C_1 in group 2; whilst C_1 and C_2 would be non-dominated (in the Pareto sense) in both groups. In this case, the winner classifier at the across-groups level would be different for the above cases (A) and (B) – i.e., the winner would be C_1 in case (A) and C_2 in case (B).

Scenario 3: Lexicographic approach at the across-groups level and homogeneous use of the Pareto approach at the within-group level

In this scenario, when two classifiers are compared by the optimiser, first, for each group of objectives, the Pareto optimiser determines whether one classifier dominates the other. Then, the lexicographic optimiser is used at the across-groups level in order to find the winner classifier.

Note that in this scenario the lexicographic approach at the across-groups level is applied to the *qualitative* results of the Pareto approach (whether or not a classifier dominates another) applied to *each group* of objectives, rather than the *numerical values* of the individual objective functions, since in this scenario the user assigns relative priorities to groups of objectives, rather than to individual objectives. That is, when comparing two classifiers, the lexicographic optimiser starts considering the highest-priority group of objectives. If the Pareto optimiser determines that one classifier dominates the other regarding the objectives in that group, then the dominating classifier is declared the winner of the lexicographic comparison across groups, since this is the highest-priority group – i.e., there is no need to determine the dominance relationships in the other lower-priority groups. If none of the classifiers dominates the other in that group of objectives, then there is a tie between the classifiers in that group, and the lexicographic (across-groups) optimiser proceeds considering the other groups of objectives, one in turn, in their priority order, until one classifier dominates the other for some group, when the dominating classifier is declared the winner of the lexicographic comparison across groups. If none of the classifiers dominates the other in any group of objectives, this overall tie would have to be broken by either selecting a classifier at random or using another criterion.

Conceptual Example for Scenarios 3 and 4: Consider a classification task where the class variable indicates whether or not an employee should be promoted. The first group, predictive accuracy measures, has two objective functions to be maximised: the Area Under the ROC curve (AUROC) and the Area Under the Precision-Recall curve (AUPRC) [25]. The user decided that neither of these two measures has priority, so a Pareto approach is appropriate for this objective group. The second group has two objective functions related to classification fairness, both to be minimised: the difference of True Positive Rates (TPR-diff) between males and females, and the difference of True Negative Rates (TNR-diff) between males and females [38]. Again, the user decided that neither of these two objectives has priority over the other, so a Pareto approach is appropriate for this objective group also. At the across-groups level, however, the user decided that the group of predictive accuracy measures has higher priority than the group of fairness measures.

Numerical Example for Scenarios 3 and 4: Consider two classifiers C_1 and C_2, whose values for each of the 4 objectives are as shown in Table 4. Regarding the two accuracy-related objectives in group 1, C_1 has a better AUROC value but a worse AUPRC value than C_2, so none of these classifiers dominates the other in objective group 1. Regarding the two fairness-related objectives in group 2, C_1 is better than C_2 regarding both TPR-diff and TNR-diff, so C_1 dominates C_2 in objective group 2. Then, comparing C_1 and C_2 across the two groups of objectives, based on the qualitative results of the Pareto-dominance check within each group, the lexicographic optimiser first checks the Pareto-dominance result for the higher-priority group 1 (accuracy measures). C_1 and C_2 are tied in group 1, since none of them dominates the other, so the lexicographic (across-groups) optimiser checks next the Pareto-dominance result for the lower-priority group 2 (fairness measures). C_1 dominates C_2 regarding the objective group 2, therefore, C_1 is the winner of the lexicographic comparison across groups.

Table 4: Example for the use of the hybrid framework in scenarios 3 and 4

Classifier	Objectives in group 1		Objectives in group 2	
	AUROC	AUPRC	TPR-diff	TNR-diff
C_1	0.73	0.60	0.20	0.25
C_2	0.70	0.64	0.22	0.30

It is worth considering also a variation of the example in Table 4 where C_2 would have an AUROC $\geq$ 0.73, and all other data in Table 4 would remain the same. In this case, C_2 would dominate C_1 regarding objective group 1. In this case, when applying the lexicographic approach across the two objective groups, C_2 would be immediately declared the overall (lexicographic) winner, due to it being the winner for the higher-priority group 1; i.e. there would be no need to check the Pareto-dominance results for the lower-priority group 2.

It is interesting to note that in this scenario there is no need to specify a tolerance threshold for the lexicographic approach, because the lexicographic optimiser is applied to the binary results of Pareto-dominance relations computed within each group of objectives, rather than applied to continuous objective values. Hence, this scenario avoids one of the aforementioned criticisms of the lexicographic approach, the need to specify ad-hoc tolerance thresholds.

Scenario 4: Lexicographic approach at the across-groups level and heterogenous use of the Pareto and lexicographic approaches at the within-group level

In this scenario the user has chosen to use the lexicographic approach at the across-group level (like Scenario 3), and has chosen to use the Pareto approach in some group(s) and the

lexicographic approach in other group(s) of objectives, at the within-group level. In our running example (Table 4), this scenario involves two different cases, as follows.

Case (A): Pareto approach in group 1 and lexicographic approach in group 2: In group 1, classifiers C_1 and C_2 are non-dominated (neither dominates the other), as mentioned earlier for Scenario 3. In group 2, regardless of which objective is chosen by the user to have higher priority, classifier C_1 wins the lexicographic comparison, since C_1 is better than C_2 regarding both objectives. Therefore, the Pareto lexicographic optimiser at the across-groups level considers that there is a tie in group 1 and proceeds to consider group 2, where C_1 is the lexicographic winner. Therefore, C_1 is the winner at the across-groups level.

Case (B): Lexicographic approach in group 1 and Pareto approach in group 2: Assume that the user has specified that AUPRC has priority over AUROC, based on the argument that AUPRC copes better with class imbalance [51], [44]; and the tolerance threshold has been set to 0.01 (as in the example for scenarios 1 and 2). In this case, classifier C_2 wins the lexicographic comparison in group 1, and therefore C_2 is also the lexicographic winner at the across-groups level, regardless of the values of the objectives in group 2 for C_1 and C_2. If, however, the used had decided that AUROC has priority over AUPRC, then C_1 would win lexicographically in group 1 and would also be the winner classifier at the across-groups level.

5. CONCLUSIONS

In real-world applications of classification (supervised learning) algorithms, particularly in high-stakes applications involving decisions about people, users often would like to optimise several quality criteria of the learned predictive models – i.e., optimising not only predictive accuracy, but also, e.g., model interpretability, fairness, privacy, etc. Despite this, the large majority of works on classification are still optimising a single objective (criterion), typically predictive accuracy. Even when multiple objectives are optimised, most works in this area use a simple weighted-sum approach, with numerical weights assigned to the objectives to be optimised, which in practice transforms the original multi-objective problem into a single-objective one (optimising the weighted sum). This simple approach is inefficient and ineffective in general [13], [11], [19]. Hence, this article focused on two genuinely multi-objective optimisation approaches which in general avoid the drawbacks of the weighted-sum approach, namely the Pareto and the lexicographic approaches.

As mentioned earlier, between these two, the Pareto approach is much more popular in machine learning. Actually, several surveys of multi-objective optimisation (MOO) do not even mention the lexicographic MOO approach [56], [57], [34], [35], [43], [54]; and so the literature often gives the misleading impression that the Pareto approach is the only good genuinely MOO approach available for researchers and practitioners. To correct that misleading impression, this article discussed the pros and cons of the Pareto and lexicographic approaches, showing that they are largely complementary; i.e., none of these two approaches is inherently better than the other. In real-world high-stakes applications, the choice between these two multi-objective optimisation approaches should be made based mainly on the needs and interests of users and the requirements of the target application domain.

In addition, this article has proposed a new conceptual, hybrid MOO framework, designed for synergistically combining the best aspects of the Pareto and lexicographic approaches. This framework provides the basis for the design of effective MOO algorithms in supervised machine learning, allowing users to flexibly decide which group(s) of objectives should be optimised according to the principles of the Pareto or lexicographic approach. This article has also given several hypothetical but plausible conceptual examples of the use of the framework, which hopefully illustrate the advantages of flexibly combining Pareto and lexicographic concepts into an MOO optimiser.

However, this article has the clear limitation of being just a position paper. Therefore, a natural direction for future research would be to design hybrid Pareto/lexicographic MOO classification (supervised learning) algorithms based on this framework, as well as empirically evaluating their effectiveness in high-stakes real-world machine learning applications.

6. ACKNOWLEDGMENTS

This work was funded by a research grant from the Leverhulme Trust, UK, reference number RPG-2020-145.

7. REFERENCES

[1] Aivodji, U., Ferry, J., Gambs, S., Huguet, M.J., Siala, M. Learning fair rule lists. *arXiv:1909.03977v1*, 9 Sep. 2019.

[2] Anahideh, H., Nezami, N., Asudeh, A. On the choice of fairness: Finding representative fairness metrics for a given context. *preprint arXiv:2109.05697*, 11 pages. 13 Sep. 2021.

[3] Barbudo, R., Ventura, S., Romero, J.R. Eight years of AutoML: categorisation, review and trends. *Knowledge and Information Systems*, *65*, 5097-5149. 2023.

[4] Bogatinovski, J., Todorovski, L., Dzeroski, S., Kocev, D. Comprehensive comparative study of multi-label classification methods. *Expert Systems with Applications*, *203*, 117215, 18 pages. 2022.

[5] Brookhouse, J. and Freitas, A. Fair feature selection: a comparison of multi-objective genetic algorithms. *arXiv preprint arXiv:2310.02752*. 2023.

[6] Burkart, N. and Huber, M.F. A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, *70*, 245-317, 2021.

[7] Cano, J.R., Gutierrez, P.A., Krawczyk, B., Woźniak, M. and Garcia, S., 2019. Monotonic classification: An overview on algorithms, performance measures and data sets. *Neurocomputing, 341,* 168-182, May 2019.

[8] Carvalho, T., Moniz, N., Faria, P., Antunes, L. Towards a data privacy-predictive performance trade-off. *Expert Systems with Applications, 223,* 119785, 1 Aug. 2023.

[9] Chouldechova, A. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, *5*(2), 153-163. 1 June 2017.

[10] Corbett-Davies, S. and Goel, S. The measure and mismeasure of fairness: A critical review of fair machine learning. *preprint arXiv:1808.00023v3*, 14 Aug. 2023.

[11] Corne, D., Deb, K., Fleming, P.J. The good of the many outweighs the good of the one: evolutionary multi-objective optimization. *IEEE Connections Newsletter 1(1),* 9-13. Feb. 2003.

[12] Crook, B., Schlüter, M., Speith, T. Revisiting the performance-explainability trade-off in explainable artificial intelligence (XAI). *preprint arXiv:2307.14239*, 2023.

[13] Deb, K. Multi-Objective Optimization Using Evolutionary Algorithms. 536 pages. Wiley, 2001.

[14] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. A fast and elitist multiobjective genetic algorithms: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6(2),* 182-197, Apr. 2002.

[15] Dong, J.D., Cheng, A.C., Juan, D.C., Wei, W., Sun, M. Dpp-net: Device-aware progressive search for pareto-optimal neural architectures. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 517-531. 2018.

[16] Elsken, T. and Hutter, F. Efficient multi-objective neural architecture search via Lamarckian evolution. In: *Proceedings of the International Conference on Learning Representations (ICLR 2019)*. *arXiv:1804.09081v4*, Feb. 2019.

[17] Emmerich, M.T.M. and Deutz, A. H. A tutorial on multiobjective optimization: fundamentals and evolutionary methods, *Natural computing*, *17(3)*, pp. 585–609, 2018.

[18] Freitas, A.A. A critical review of multi-objective optimization in data mining: a position paper. *ACM SIGKDD Explorations, 6(2)*, pp. 77-86. ACM, Dec. 2004.

[19] Gardner, S., Golovidov, O., Griffin, J., Koch, P., Thompson, W., Wujek, B., Xu, Y. Constrained multi-objective optimization for automated machine learning. In: *Proceedings of the 2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA)* (pp. 364-373). IEEE, 2019.

[20] Gonzales, J., Ortego, J., Escobar, J.J., Damas, M. A lexicographic cooperative co-evolutionary approach for feature selection. *Neurocomputing, 463,* 59-76, 6 Nov. 2021.

[21] Grandini, M., Bagli, E., Visani, G. Metrics for multi-class classification: an overview. *preprint arXiv:2008.05756*, 2020.

[22] Gutierrez, P.A. and García, S. Current prospects on ordinal and monotonic classification. *Progress in Artificial Intelligence*, *5*(3), 171-179, Aug. 2016.

[23] Hand, D. Measuring classifier performance: a coherent alternative to the area under the ROC curve. *Machine Learning, 77*, 103-123, 2009.

[24] Hong, M.F., Chen, H.Y., Chen, M.H., Xu, Y.S., Kuo, H.K., Tsai, Y.M., Chen, H.J., Jou, K. Network Space Search for Pareto-Efficient Spaces. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3053-3062, 2021.

[25] Japkowicz, N. and Shah, M. *Evaluating Learning Algorithms.* Cambridge University Press, 2011.

[26] Karmaker, S.K., Hassan, M., Smith, M.J., Xu, L., Zhai, C. AutoML to date and beyond: challenges and opportunities. *ACM Computing Surveys, 54(8),* Article 175, Oct. 2021.

[27] Kearns, M., Neel, S., Roth, A., Wu, Z.W. An empirical study of rich subgroup fairness for machine learning. In: *Proceedings of the conference on Fairness, Accountability and Transparency (FAT'19)*, 100-109. 2019.

[28] Kim, Y.H., Reddy, B., Yun, S., Seo, C. NEMO: neuro-evolution with multiobjective optimization of deep neural network for speed and accuracy. *JMLR: Workshop and Conference Proceedings 1:* 1-8, 2017.

[29] Kleinberg, J., Mullainathan, S., Raghavan, M. Inherent trade-offs in the fair determination of risk scores. *preprint arXiv:1609.05807v2*, 23 pages. 17 Nov. 2016.

[30] Kuhn, M., Johnson, K. *Applied Predictive Modeling*. Springer, 2013.

[31] Li, X., Zhou, Y., Pan, Z., Feng, J. Partial order pruning: for best speed/accuracy trade-off in neural architecture search. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9145-9153. CFV, 2019.

[32] Li, W., Wang, R., Zhang, T., Ming, M., Li, K. Reinvestigation of evolutionary many-objective optimization: focus on the Pareto knee front. *Information Sciences*, *522*, 193-213, 2020.

[33] Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li, Y., Liu, X., He, B. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering, 35(4)*, 3347-3366, April 2023.

[34] Liang, J., Ban, X., Yu, K., Qu, B., Qiao, K., Yue, C., Chen, K., Tan, K.C.. A survey on evolutionary constrained multiobjective optimization. *IEEE Transactions on Evolutionary Computation, 27(2)*, 201-221, April 2023.

[35] Liu, S., Lin, Q., Li, J., Tan, K.C. A survey on learnable evolutionary algorithms for scalable multiobjective optimization. *IEEE Transactions on Evolutionary Computation, 27(6)*, 1941-1961. Dec. 2023.

[36] Lu, Z., Whalen, I., Boddeti, V., Dhebar, Y., Deb, K., Goodman, E., Banzhaf, W. NSGA-net: neural architecture search using multi-objective genetic algorithm. In: *Proceedings of the 2019 Genetic and Evolutionary Computation Conference (GECCO)*, 419–427. ACM, 2019.

[37] Malley, J.D., Malley, K.G., Pajevic, S. *Statistical learning for biomedical data*. Cambridge University Press, 2011.

[38] Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A. A survey on bias and fairness in machine learning. *ACM Computing Surveys, 54, 6*, Article 115, July 2021.

[39] Mitchell, T. *Machine Learning*. McGrow-Hilll, 1997.

[40] Mittelstadt, B., Russell, C., Wachter, S. Explaining explanations in AI. In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 279-288. ACM 2019.

[41] Molnar, C., Konig, G., Herbinger, J., Freiesleben, T., Dandl, S., Scholbeck, C.A., Casalicchio, G., Grosse-Wentrup, M., Bischl, B. General pitfalls of model-agnostic interpretation methods for machine learning models. *preprint arXiv:2007.04131v2*, *17 Aug. 2021*.

[42] Monteiro, W.R., Reynoso-Meza, G. A review of the convergence between explainable artificial intelligence and multi-objective optimization. *Pre-print at Techrxiv.org*, 2022.

[43] Morales-Hernandez, A., Van Nieuwenhuyse, I., Gonzales, S.R. A survey on multi-objective hyperparameter optimization algorithms for machine learning. *Artificial Intelligence Review, 56*: 8043-8093. 2023.

[44] Movahedi, F., Padman, R., Antaki, J.F. Limitations of ROC on imbalanced data: Evaluation of LVAD mortality risk scores. *preprint arXiv:2010.1625*, 2020.

[45] Pereira, R.B., Plastino, A., Zadrozny, B., Mershmann, L.H.C. Correlation analysis of performance measures for multi-label classification. *Information Processing and Management, 54,* 359-369, 2018.

[46] Petchrompo, S., Coit, D.W., Brintrup, A., Wannakrairot, A., Parlikad, A.K. A review of Pareto pruning methods for multi-objective optimization. *Computers & Industrial Engineering, 167*, 108022, May 2022.

[47] Pfisterer, F., Coors, S., Thomas, J., Bischl, B. Multi-objective automatic machine learning with AutoxgboostMC. *arXiv pre-print: arXiv:1908.10796v2,* 30 Apr. 2021.

[48] Poyiadzi, R., Renard, X., Laugel, T., Santos-Rodriguez, R., Detyniecki, M. Understanding surrogate explanations: the

interplay between complexity, fidelity and coverage. *preprint arXiv:2107.04309*. 9 July 2021.

[49] Quadrianto, N., Sharmanska, V. Recycling privileged learning and distributed matching for fairness. In: *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, 677-688, 2017.

[50] Rezaei, S., Shafiq, Z., Liu, X. Accuracy-privacy trade-off in deep ensemble: a membership inference perspective. In: *Proceedings of the 2023 IEEE Symposium on Security and Privacy (SP)*, 364-381. IEEE, 2023.

[51] T. Saito, M. Rehmsmeier. The Precision-Recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS One*, March 4, 2015, 21 pages.

[52] L. Schneider, B. Bischl, J. Thomas. Multi-objective optimization of performance and interpretability of tabular supervised machine learning models. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-23)*, 538-547. ACM Press, 2023.

[53] Sener, O. and Koltun, V. Multi-task learning as multi-objective optimization. *Advances in Neural Information Processing Systems*, *31*, 2018.

[54] Sharma, S., Kumar, V. A comprehensive review on multi-objective optimization techniques: past, present and future. *Archives of Computational Methods in Engineering, 29*, 5605-5633, 2022.

[55] Sudeng, S. and Wattanapongsakorn, N., Pruning algorithm for Multi-objective optimization. In: *Proceedings of the 10th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 70-75. IEEE, 2013.

[56] Taha, K. Methods that optimize multi-objective problems: a survey and experimental evaluation. *IEEE Access, 8*, 80855-80878, 2020.

[57] Tian, Y., Si, L., Zhang, X., Cheng, R., He, C., Tan, K.C., Jin, Y. Evolutionary large-scale multi-objective optimization: a survey. *ACM Computing Surveys*, *54*(8), Article 174, Oct. 2021.

[58] Tsoumakas, G., Katakis, I., Vlahavas, I. Mining multi-label data. In: O. Maimon and L. Rokach (Eds.) *Data Mining and Knowledge Discovery Handbook*, 2nd Ed. Springer, 2010.

[59] Valdivia, A., Sanchez-Monedero, J., Casillas, J. How fair can we go in machine learning? Assessing the boundaries of accuracy and fairness. *Int. J. Intelligent Systems, 36(4), 2021*, 1619-1643.

[60] Verma, S. and Rubin, J. Fairness definitions explained. In: *Proceedings of the 2018 IEEE/ACM International Workshop on Software Fairness (FairWare)*, 1-7. IEEE, 2018.

[61] Wang, H., Olhofer, M., Jin, Y. A mini-review on preference modeling and articulation in multi-objective optimization: current status and challenges. *Complex & Intelligent Systems*, *3*, 233-245, 2017.

[62] Wang, S., Wang, Y., Wang, D., Yin, Y., Wang, Y., Jin, Y. (2020). An improved random forest-based rule extraction method for breast cancer diagnosis. *Applied Soft Computing*, *86*, 105941.

[63] Wei, S., Niethammer, M.. The fairness-accuracy Pareto front. *Statistical Analysis and Data Mining*, *15(3)*, 287-302, June 2022.

[64] Witten, I.H., Frank, E., Hall, M.A., Pal, C.J. Data Mining: practical machine learning tools and techniques. 4th Ed. Morgan Kaufmann, 2016.

[65] Yang, Y., Nam, A., Nasr-Azadani, M., Tung, T.. Resource-aware pareto-optimal automated machine learning platform. In: *Proceedings of the 2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 6 pages. IEEE, 2020.

[66] Zhu, H., Zhang, H., Jin, Y. From federated learning to federated neural architecture search: a survey. *Complex & Intelligent Systems*, *7*(2), pp.639-657, 2021.

[67] Zoller, M.A. and Huber, M.F. Benchmark and survey of automated machine learning frameworks. *Journal of Artificial Intelligence Research 70*, 409-472, 2021.

[68] Zhang, S., Jia, F., Wang, C., Wu, Q. Targeted hyperparameter optimization with lexicographic preferences over multiple objectives. In: *Proceedings of the Eleventh International Conference on Learning Representations (ICLR 2023)*, 16 pages.

About the author:

Alex A. Freitas is a Professor of Computational Intelligence and currently the Head of Research at the School of Computing, University of Kent, UK. He has an interdisciplinary academic background, with a PhD in Computer Science from the University of Essex, UK (1997); and an MPhil (a research-oriented master's degree) in Biological Sciences from the University of Liverpool, UK (2011). His main research interests are classification (supervised machine learning) methods, including the issues of interpretability and fairness of classification models, as well as the applications of supervised machine learning methods to the life sciences, particularly the biology of ageing.