

Fair Bayesian Optimization

Valerio Perrone
Amazon Web Services
vperrone@amazon.com

Michele Donini
Amazon Web Services
donini@amazon.com

Muhammad Bilal Zafar
Amazon Web Services
zafamuh@amazon.com

Robin Schmucker*
Carnegie Mellon University
rschmuck@andrew.cmu.edu

Krishnaram Kenthapadi
Amazon Web Services
kenthk@amazon.com

Cédric Archambeau
Amazon Web Services
cedrica@amazon.com

ABSTRACT

Given the increasing importance of machine learning (ML) in our lives, several algorithmic fairness techniques have been proposed to mitigate biases in the outcomes of the ML models. However, most of these techniques are specialized to cater to a single family of ML models and a specific definition of fairness, limiting their adaptability in practice. We introduce a general constrained Bayesian optimization (BO) framework to optimize the performance of any ML model while enforcing one or multiple fairness constraints. BO is a model-agnostic optimization method that has been successfully applied to automatically tune the hyperparameters of ML models. We apply BO with fairness constraints to a range of popular models, including random forests, gradient boosting, and neural networks, showing that we can obtain accurate and fair solutions by acting solely on the hyperparameters. We also show empirically that our approach is competitive with specialized techniques that enforce model-specific fairness constraints, and outperforms preprocessing methods that learn fair representations of the input data. Moreover, our method can be used in synergy with such specialized fairness techniques to tune their hyperparameters. Finally, we study the relationship between fairness and the hyperparameters selected by BO. We observe a correlation between regularization and unbiased models, explaining why acting on the hyperparameters leads to ML models that generalize well and are fair.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**.

KEYWORDS

AutoML, Bayesian optimization, hyperparameter tuning, fairness, bias

ACM Reference Format:

Valerio Perrone, Michele Donini, Muhammad Bilal Zafar, Robin Schmucker, Krishnaram Kenthapadi, and Cédric Archambeau. 2021. Fair Bayesian Optimization. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society (AIES '21)*, May 19–21, 2021, Virtual Event, USA. ACM, New York, NY, USA, 14 pages. <https://doi.org/10.1145/3461702.3462629>

*Work done as part of an internship at Amazon Web Services.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

AIES '21, May 19–21, 2021, Virtual Event, USA.
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8473-5/21/05.
<https://doi.org/10.1145/3461702.3462629>

1 INTRODUCTION

With the increasing use of machine learning (ML) in domains such as financial lending, hiring, criminal justice, and college admissions, there has been a major concern for the potential for ML to unintentionally encode societal biases and result in systematic discrimination [4–6, 8, 10]. For example, a classifier that is only tuned to maximize prediction accuracy may unfairly predict a high credit risk for some subgroups of the population applying for a loan. Extensive work has been done to measure and mitigate biases during different stages of the ML life-cycle [5].

Specifically, a number of methods have been proposed in recent years to reduce unfairness in the ML model outcomes. The key design principle behind all of these fairness interventions is to maximize prediction accuracy subject to some constraints on the fairness of the outcomes *while preserving computational tractability*.¹ However, as an artefact of the need to preserve computational tractability, these fairness interventions suffer from one or more of the following drawbacks. The fairness intervention is (i) specific to the model class (e.g., linear models only), (ii) limited to a specific definition(s) of fairness, (iii) limited to a single, binary sensitive feature, (iv) requires access to sensitive feature information at prediction time, or (v) results in a randomized classifier that may generate different prediction for the same input at different times. These drawbacks limit the practitioners' ability to deploy fair ML solutions for arbitrary machine learning pipelines corresponding to different practical goals and constraints.² For instance, as a result of limitation (i), any ensemble or hybrid learning solutions combining different model classes and/or domain knowledge are ruled out. As a drawback of limitation (iv), the corresponding fairness intervention cannot be used when the sensitive feature information is not available at prediction time, while limitation (v) may result in the same model accepting and denying the loan to the same applicant at different times.

In this work, we draw upon the insight that hyperparameter optimization techniques have been used successfully in real-world to optimize the traditional objective of accuracy for arbitrary learning pipelines. For instance, several cloud platforms allow users to bring their data along with their own unknown and proprietary model accompanied by any kind of auxiliary pre/post-processing steps; perform model training; and tune the hyperparameters; *all while*

¹Computational tractability here refers to learning or implementing the fairness intervention in a time-efficient manner.

²A pipeline here refers to a ML model (e.g., DNN) with custom input pre-processing and output post-processing steps attached to it.

treating the pipeline as an opaque box whose internals cannot be modified [30, 49].³

Motivated by the versatility of hyperparameter tuning, we present a general constrained Bayesian Optimization (BO) framework to tune the performance of ML models with constraints on fairness. In BO, the desired objective is described through a probabilistic model, which not only predicts the best estimates (posterior means), but also uncertainties (posterior variances) for each hyperparameter configuration. The next configuration to try strikes a balance between exploration (sampling where the objective is most uncertain) and exploitation (minimizing the best estimate) [54]. As a result, BO finds promising hyperparameter configurations more efficiently than random search. Introductions to Bayesian optimization are provided in [7] and [52].

Our framework is agnostic to the type of ML model and its training procedure. Hence, it can be applied to arbitrary learning pipelines. We demonstrate its effectiveness on several classes of ML models, including random forests, gradient boosting, and neural networks, showing that we can obtain accurate and fair models simply by acting on their hyperparameters. Figure 1 illustrates this idea by plotting the accuracy and unfairness levels achieved by trained gradient boosted tree ensembles (XGBoost [16]), random forests (RF) and fully-connected feed-forward neural networks (NN), with each dot corresponding to a random hyperparameter configuration. The key observation is that, given a level of accuracy, one can reduce unfairness just by tuning the model hyperparameters. As an example, paying 0.02 points of accuracy (-2.5%, from 0.69 to 0.67) can give a RF model with 0.08 fewer unfairness points (-70%, from 0.13 to 0.04).

It is not uncommon to have laws and regulations enforcing a well-defined fairness metric or ranges of acceptable fairness levels (e.g., the 80% rule for Disparate Impact in [24]). In these settings, the constrained BO framework we propose is a more suitable choice than multi-objective optimization techniques [42, 46], which would explore the entire Pareto front of feasible and accurate solutions irrespective of the required fairness level. Further experiments show that our model-agnostic approach is also more effective than pre-processing techniques that learn “fair representations” of the input data, and is competitive with model-specific methods that have access to the model internals and incorporate fairness constraints as part of the objective during model training.

The paper is organized as follows. Section 2 presents the main ideas of algorithmic fairness, such as state-of-the-art methods and common definitions of fairness. Section 3 introduces our model-agnostic methodology to optimize model hyperparameters while satisfying fairness constraints. The experimental results in Section 4 show that our method compares favourably with state-of-the-art techniques. We also analyze the importance of hyperparameters controlling regularization in the search for fair and accurate models. Finally, Section 5 presents conclusions and future directions.

We open-sourced our code for constrained BO through AutoGluon [23], which is available at the following link: <https://github.com/awslabs/autogluon/>.

³SigOpt (<https://sigopt.com/>); Cloud AutoML (<https://www.blog.google/products/google-cloud/cloud-automl-making-ai-accessible-every-business/>); Optuna (<https://optuna.org/>); Amazon AMT (<https://docs.aws.amazon.com/sagemaker/latest/dg/automatic-model-tuning.html>).

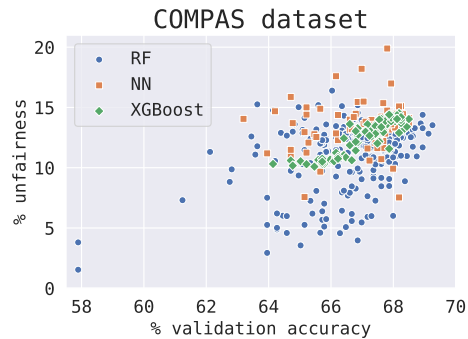


Figure 1: Unfairness-accuracy trade-off by varying the hyperparameters of XGBoost, RF, and NN on a recidivism prediction task. Each dot corresponds to a model trained with a different hyperparameter configuration. For a given level of accuracy, models with very different levels of unfairness can be generated simply by changing the model hyperparameters.

2 BACKGROUND ON ALGORITHMIC FAIRNESS

Algorithmic fairness literature focuses on developing machine learning methods that are both accurate and fair. There is extensive work on identifying and measuring the extent of discrimination (e.g., [4, 10]), and on mitigation approaches in the form of fairness-aware learning methods (e.g., [9, 12, 20, 22, 26, 27, 31, 35, 38, 58, 60, 62, 63]).

Today, there is no consensus on a unique definition of fairness and different definitions encode different values that are appropriate in distinct contexts. Moreover, some of the most common definitions are conflicting [41, 56]. Our goal is not to introduce yet another fairness definition, but to propose a flexible methodology that is able to output fair models regardless of the selected criterion we want to enforce. As we will show, in our model-agnostic framework we can seamlessly incorporate different definitions, either separately, or simultaneously.

2.1 Fairness Definitions

Let Y be the true label (binary), S the protected (or sensitive) attribute (binary), and \hat{Y} the predicted label. The most common definitions can be grouped into three categories: (i) considering the predicted outcome given the true label; (ii) considering the true label given predicted outcome; (iii) considering the predicted outcome only. The following are examples of the most used definitions.

- Equal Opportunity (EO) requires equal True Positive Rates (TPR) across subgroups: $P(\hat{Y} = 1|Y = 1, S = 0) = P(\hat{Y} = 1|Y = 1, S = 1)$;
- Equalized Odds (EOdd) requires equal False Positive Rates (FPR), in addition to EO;
- Statistical Parity (SP) requires positive predictions to be unaffected by the value of the protected attribute, regardless of the true label $P(\hat{Y} = 1|S = 0) = P(\hat{Y} = 1|S = 1)$.

Our goal is to find accurate models with a controlled (small) violation of a pre-defined fairness constraint. Hence, following [20],

we consider the family of ϵ -fair models: a model \hat{Y} is ϵ -fair if it violates the fairness definition by at most $\epsilon \geq 0$. In the case of EO, a model \hat{Y} is ϵ -fair if the difference in EO (DEO) is at most ϵ :

$$|P(\hat{Y} = 1|Y = 1, S = 0) - P(\hat{Y} = 1|Y = 1, S = 1)| \leq \epsilon. \quad (1)$$

For EO_{odd}, we have two different constraints simultaneously: the first one is equivalent to DEO and the second one is the difference of FPR (DFP). Finally, we can similarly define the difference in SP (DSP):

$$|P(\hat{Y} = 1|S = 0) - P(\hat{Y} = 1|S = 1)| \leq \epsilon. \quad (2)$$

The inherent trade-offs underlying different notions of fairness has been studied extensively [22, 26, 41]. Picking the correct definition for the problem at hand is difficult, and it cannot be delegated to an automatic agent in practice. Instead, a human decision is preferred to ensure an informed decision (e.g., with human-in-the-loop approaches [59]).

2.2 Mitigating unfairness

Unfairness mitigation techniques can be divided into pre-, in- and post-processing techniques.

Methods in the first family achieve fairness by modifying the data representation, i.e., **pre-processing** the data, and then to adopt standard ML methods [11, 37, 62]. For instance, [62] learn a fair representation by solving an optimization problem with a two-fold goal: encode the data by preserving as much information as possible and obfuscate the sensitive feature group membership. Whereas techniques like that of [37] remove the sensitive attribute from the feature set, and rebalance the dataset using SMOTE-like procedures [15] to under- or over-sample observations from certain sensitive feature groups. Pre-processing methods are model-agnostic, but their hyperparameters, as well as the ones of the selected ML model still need to be tuned for performance. As we will show shortly, this can negatively impact the fairness and accuracy of the returned solution.

The second family consists of **in-processing** methods that enforce model-specific fairness constraints during training (e.g., [20, 60, 61]). These methods are, however, only applicable to a certain model class. For example, the algorithm proposed in [20] can be applied only to kernel machines (such as support vector machines), and only for a single definition of fairness (i.e., EO). Although in-processing methods may perform well for the model class that they were designed for, they are often difficult, or sometimes impossible to extend to new model classes. These methods may also introduce new hyperparameters [61], tuning which may require further domain expertise and/or validation.

Post-processing techniques operate by adjusting the decision threshold of a pre-trained classifier to make the eventual outcomes fairer with respect to a given fairness definition. One of the most popular methods in this family is [31], which also introduces the concept of EO. The main drawback of these techniques is that post-processing the decision threshold is inherently sub-optimal and may lead to worse fairness-accuracy tradeoffs [58]. Moreover, these techniques are not usable when the sensitive feature information is not available at prediction time, and might be prone to legal challenges due to the use of sensitive feature for making predictions [43].

Method	(i)	(ii)	(iii)	(iv)	(v)
Zemel et al. [62]	✓	✗	✗	✓	✓
Kamiran et al. [37]	✓	✗	✗	✓	✓
Donini et al. [20]	✗	✗	✓	✓	✓
Zafar et al. [60, 61]	✗	✗	✓	✓	✓
Hardt et al. [31]	✓	✗	✗	✗	✗
Agarwal et al. [2]	✓	✓	✓	✓	✗
Kearns et al. [40]	✓	✓	✓	✓	✗
Ours	✓	✓	✓	✓	✓

Table 1: Capabilities of different unfairness mitigation methods w.r.t. desirable practical properties outlined in Section 1: (i) Method caters to arbitrary model classes, (ii) arbitrary fairness definitions, (iii) multiple/polyvalent sensitive features, (iv) can operate without access to sensitive feature at prediction time, and (v) outputs non-randomized predictions.

A recently proposed class of methods, referred to as **meta algorithms**, reduce the fair classification task to a sequence of cost-sensitive classification problems [2, 3, 40]. The solutions to these problems yield a randomized classifier. As opposed to in-processing methods, the meta algorithms do not depend on the model class of the underlying classifier, rather on the ability to retrain it repeatedly. The connection between randomized classifiers and fairness (and consequently the connection to differential privacy) has also been studied [45]. In the context of empirical risk minimization algorithms, these methods are model-agnostic with respect to the ML model, but still need specific implementations based on the selected fairness definition and require to output an ensemble of models.⁴ Similar limitations characterize a number of approaches that either use optimization [40, 55] or Bayesian inference [17, 19, 25]. Their implementation has to be tailored to the specific fairness definitions or cannot be applied out-of-the-box to arbitrary ML models and training algorithms. In contrast, we show that our constrained BO approach is agnostic, both, to the selected ML model and to the desired fairness constraint (or ensemble of constraints).

Table 1 summarizes the the existing methods in terms of the capabilities mentioned in Section 1. The table shows that none of the existing methods satisfies all of the desirable practical properties.

Finally, as many existing approaches discussed in this section require tuning certain hyperparameters, we note that our proposal can be used in synergy with these methods to tune their hyperparameters.

3 FAIR BAYESIAN OPTIMIZATION

Bayesian optimization (BO) is a well-established methodology to optimize expensive opaque functions (see [52] for an overview). It relies on a probabilistic model of the unknown target $f(\mathbf{x})$ one wishes to optimize. The opaque function $f(\mathbf{x})$ is repeatedly queried until one runs out of budget (e.g., time). Queries consist of evaluations of f at hyperparameter configurations $\mathbf{x}^1, \dots, \mathbf{x}^n$ selected according to an explore-exploit trade-off criterion or *acquisition function* [36]. The hyperparameter configuration corresponding to

⁴Fairlearn code at: <https://github.com/fairlearn/fairlearn>.

the best query is then returned. A popular approach is to impose a Gaussian process (GP) prior over f and then compute the posterior GP based on the observed queries $f(\mathbf{x}^1), \dots, f(\mathbf{x}^n)$ [51]. The posterior GP is characterized by a posterior mean function and a posterior variance function that are required when evaluating the acquisition function for each new query of f .

A widely used acquisition function is the Expected Improvement (EI) [44]. This is defined as the expected amount of improvement of an evaluation with respect to the current minimum $f(\mathbf{x}_{min})$. For a Gaussian predictive distribution, EI is defined in closed-form as

$$\begin{aligned} EI(\mathbf{x}) &= \mathbf{E}[\max(0, f(\mathbf{x}_{min}) - f(\mathbf{x}))] \\ &= \sigma^2(\mathbf{x})(z(\mathbf{x})\Phi_n(z(\mathbf{x})) + \phi_n(z(\mathbf{x}))), \end{aligned}$$

where $z(\mathbf{x}) := \frac{\mu(\mathbf{x}) - f(\mathbf{x}_{min})}{\sigma(\mathbf{x})}$. Here, μ and σ^2 are respectively the posterior GP mean and variance, and Φ_n and ϕ_n respectively the CDF and PDF of the standard normal. Alternative acquisition functions based on information gain criteria have also been developed [32, 33, 57]. Standard acquisitions focus only on the objective $f(\mathbf{x})$ and do not account for additional constraints. In this work, we aim to optimize an opaque function $f(\mathbf{x})$ subject to fairness constraints $c(\mathbf{x}) \leq \epsilon$, with $\epsilon \in \mathbb{R}^+$ determining how strictly the corresponding fairness definition should be enforced.

We next describe Fair Bayesian Optimization (FairBO), our approach to optimize the hyperparameters of an opaque function in a model-agnostic manner while satisfying arbitrary fairness constraints (see Algorithm 1). For simplicity, we consider the constrained EI (cEI), an established acquisition function to extend BO to the constrained case [28, 29, 53]. Alternatively, FairBO could straightforwardly leverage entropy-based acquisition functions [33, 48]. We model the fairness level $c(\mathbf{x})$ via an additional GP and weight the EI with the posterior probability of the constraint being satisfied. This gives $cEI(\mathbf{x}) = EI(\mathbf{x})P(c(\mathbf{x}) \leq \epsilon)$, where the probability of satisfying the fairness constraint $P(c(\mathbf{x}) \leq \epsilon)$ is computed by evaluating the univariate Gaussian cumulative distribution. In our setting, the feasible hyperparameter configurations are those satisfying the desired fairness constraint (e.g., the DSP across subgroups should be lower than 0.1). The observed feedback on the fairness metric may also be binary in that, rather than the exact fairness value, only a binary signal in $\{-1, +1\}$ may be observed when the constraint is respectively violated or satisfied. In this case, $c(\mathbf{x})$ can be modelled through a GP classification model with a Bernoulli likelihood, with the probability of satisfying the constraint being the posterior probability of the positive class [29, 51].

We define $cEI(\mathbf{x})$ as the expected improvement with respect to the current *fair* best, namely the hyperparameter configuration with the best performance metric that satisfies the fairness constraint. As this may not be available in the initial iterations, we start by greedily optimizing $P(c(\mathbf{x}) \leq \epsilon)$ and then switch to $cEI(\mathbf{x})$ (lines 4-10) when the first fair hyperparameter configuration is found. Following the approach from [28] and [29], this ensures that BO maximizes the probability of finding a feasible configuration before starting to account for performance metrics, which is particularly relevant when the region of feasible hyperparameter configurations is small. FairBO is straightforward to extend to handle D fairness constraints simultaneously, each with its own upper bound $\{\epsilon_d\}_{d=1}^D$. We merge the fairness constraints into a single binary feedback

encoding whether all constraints are satisfied. Assuming independence, one can alternatively place a fairness model $\{c_d(\mathbf{x})\}_{d=1}^D$ on every fairness constraint and let $P(\mathbf{x}) = \prod_{d=1}^D P(c_d(\mathbf{x}) \leq \epsilon_d)$, each term being the probability of satisfying a fairness constraint.

FairBO assumes the user specifies the desired fairness constraints *a priori*. As a result, the method can explore the region of the hyperparameter space that satisfies the fairness constraints in a data-efficient way. When the user does not specify the desired fairness constraints, one would be forced to rely on a multi-objective strategy to recover the Pareto front of fair and accurate solutions, selecting an acceptable fairness and accuracy level *a posteriori* [13, 14, 50]. Multi-objective optimization comes with its own challenges (e.g., how to trade-off multiple objectives). As we will see shortly, it is also inherently more expensive computationally and slower than FairBO to return a solution as it does not focus on the feasible region.

Algorithm 1 FairBO

Input: Initial and total budgets T_0, T ; unfairness bound ϵ ; GP prior on objective $f(\mathbf{x})$ and fairness $c(\mathbf{x})$ model.

- 1: Evaluate $f(\mathbf{x}_i)$ and $c(\mathbf{x}_i)$ for $i = 1, \dots, T_0$ hyperparameters \mathbf{x}_i from the search space (e.g., drawn uniformly at random or from a fixed initial design) and set the used budget $t = T_0$.
 - 2: Define the set of evaluated hyperparameters $C = \{(\mathbf{x}_i, f(\mathbf{x}_i), c(\mathbf{x}_i))\}_{i=1}^{T_0}$
 - 3: Compute the posterior GP for the objective and the fairness models based on C .
 - 4: **while** $t < T$ **do**
 - 5: $\mathbf{x}_{new} \in \arg \max_{\mathbf{x}} EI(\mathbf{x})P(c(\mathbf{x}) \leq \epsilon)$.
 - 6: Evaluate $f(\mathbf{x}_{new})$ and $c(\mathbf{x}_{new})$.
 - 7: Update $C = C \cup \{(\mathbf{x}_{new}, f(\mathbf{x}_{new}), c(\mathbf{x}_{new}))\}$
 - 8: Compute the posterior GP for the objective and the fairness models based on C .
 - 9: $t = t + 1$
 - 10: **end while**
 - 11: **return** Best fair hyperparameter configuration in C .
-

4 EXPERIMENTS

We demonstrate the benefits of FairBO through extensive experiments. Our evaluation is driven by four key messages. First, we show that FairBO is an efficient strategy to find well-performing and fair hyperparameter configurations, converging to a good solution faster than standard BO and random search. Second, we show that FairBO is agnostic to the fairness constraint and can naturally handle multiple constraints simultaneously. Third, we shed light on the relationship between hyperparameters and fairness metrics, finding a connection between regularization hyperparameters and fairness. Finally, we show that FairBO compares favourably to state-of-the-art model-agnostic and model-specific fairness techniques.

We consider three datasets widely used in the context of fairness: (i) Adult – Census Income [21], a binary classification problem with binary gender as sensitive attribute, where the goal is to predict if income exceeds \$50K/yr based on census data; (ii) German Credit Data [21], a binary classification problem with binary gender as

sensitive attribute, where the goal is to classify people described by a set of attributes as good or bad credit risks; (iii) COMPAS, a binary classification problem concerning recidivism risk, with binarized ethnic group as sensitive attribute (one group for “white” and one for all other ethnic groups).⁵ We tune four popular ML algorithms implemented in `scikit-learn` [47]: XGBoost, Random Forest (RF), a fully-connected neural network (NN), and Linear Learner (LL), optimizing the hyperparameters in Appendix A. We optimize for validation accuracy, with a random 70%/30% split into train/validation, and place an upper bound on unfairness (e.g., defined via DSP as per inequality (2)). All hyperparameter optimization methods are initialized with 5 random hyperparameter configurations. BO and FairBO are implemented in GPyOpt [1], with the GP using a Matérn-5/2 covariance kernel with automatic relevance determination hyperparameters, optimized by type-II maximum likelihood [51]. Results are averaged across 10 repetitions, with 95% confidence intervals obtained via bootstrapping. All experiments are run on AWS with `m4.xlarge` machines.

4.1 FairBO performance

We first compare FairBO to Random Search (RS) and standard BO (based on the EI acquisition). Figure 2 compares the validation error of the fair solution found on Adult, German, and COMPAS with a fairness constraint of $\text{DSP} \leq 0.05$ (i.e., the selected model has a difference in probability of predicting a positive output between the two subgroups that is less or equal than 5%, see Eq. 2 with $\epsilon = 0.05$). For each seed, all methods are initialized with the same set of five hyperparameter evaluations drawn uniformly at random from the search space. While all methods are eventually expected to find similar solutions with enough budget, we are interested in how fast they find a well-performing fair hyperparameter configuration. As expected, FairBO finds an accurate and fair model more quickly than RS and BO. On Adult, FairBO reaches the fair (local) optimum five times faster than RS. Figure 3 shows an example run with all tried hyperparameter configurations. Standard BO can get stuck in high-performing yet unfair regions, failing to return a feasible solution. While RS is more robust in that it cannot get stuck exploring unfeasible regions, it only finds a fair solution with the same accuracy of the trivial model always predicting the majority class (i.e., the set of points with accuracy ≈ 0.763). In contrast, FairBO is able to sample the fairness-performance space much more effectively. RS samples the space of hyperparameters uniformly, which does not correspond to sampling the fairness-performance space uniformly, while standard BO does not receive any signal from violating the constraint and blindly exploits well-performing yet unfair hyperparameters.

Analogous results are shown in Appendix B for the problem of tuning XGBoost and NN. We also repeat these experiments with a different instantiation of FairBO through constrained max-value entropy-search (cMES), the state-of-the-art acquisition function for constrained BO based on information gain criteria [48]. We show

⁵COMPAS link: <https://github.com/propublica/compas-analysis>. We are aware of the ethics issues concerning COMPAS dataset. We think that recidivism prediction is itself a highly contested area. We decided to use this data only due to its popularity as benchmark dataset in the field, but our goal is not to normalize the definition of social fairness for recidivism prediction as something that can be achieved by only relying on a statistical fairness criterion *as is* without a diligent human intervention in the decision process.

that the two instantiations of FairBO achieve similar performance on these benchmarks. Appendix B also includes experiments with a looser fairness constraint $\text{DSP} \leq 0.15$. As expected, the accuracy of the best fair solution is higher under a looser constraint and FairBO still compares favorably to the baselines. In the appendix we also repeat the experiments with a constraint on DEO (inequality (1)) and confirm the benefits of FairBO, noting that other fairness definitions can be used if desired.

4.2 Multiple fairness constraints

In contrast to most algorithmic fairness techniques, FairBO can seamlessly handle multiple fairness definitions simultaneously. We consider 100 iterations of standard BO and FairBO on the problem of tuning RF on Adult and repeat the experiment three times, each time introducing an additional fairness constraint. Specifically, we first impose a constraint on DFP, then on both DFP and DEO, and finally on DFP, DEO, and DSP together. All constraint thresholds are set to 0.05 and results are averaged over 10 independent repetitions. Figure 4 shows accuracy and three fairness metrics (i.e., one minus unfairness, namely, $(1 - \text{DFP})$, $(1 - \text{DEO})$, $(1 - \text{DSP})$ respectively) of the returned fair solution for RF. Analogous results for XGBoost, NN, and LL are given in Appendix B. Interestingly, FairBO allows us to trade off relatively little accuracy for a fairer solution, which gets progressively fairer as we add more constraints.

4.3 Hyperparameters and fairness

We showed that hyperparameter tuning can mitigate unfairness effectively. We now investigate more closely the role of each hyperparameter on the unfairness of the resulting model. For each algorithm, we apply fANOVA [34] to study hyperparameter importance on fairness, defined as DSP (analogous results with DEO are given in Appendix B). Hyperparameter configurations and unfairness metrics are collected from 100 iterations of random search and 10 random seeds, for a total of 1000 data points per algorithm-dataset pair. Figure 5 indicates that the hyperparameters controlling the regularization level tend to have the largest impact on fairness. In the case of RF, the most important hyperparameter is the maximum tree depth; for XGBoost, this is either the L1 weight regularizer α or the number of boosting rounds; for NN, Adam’s initial learning rate eps plays the biggest role (as we keep the number of epochs fixed, Adam’s initial learning rate works indeed as regularizer). Finally, for LL the most relevant hyperparameter influencing fairness is precisely the regularization factor α . Figure 6 shows the DSP and accuracy for 100 random hyperparameter configurations for each algorithm, before and after fixing the most relevant hyperparameter detected by fANOVA to the default value from `scikit-learn`. As expected, freezing these hyperparameters limits the ability of FairBO to provide fair and accurate solutions, leading to fewer fair solutions.

We conjecture that, by preventing overfitting, the hyperparameters controlling regularization generate models with a lower ability to discriminate among the different values of the sensitive attribute. For example, consider the simple case in which the sensitive feature is uncorrelated with the other features. Assuming we have a linear model w , where the entry $w_s \in \mathbb{R}$ is the weight assigned to the

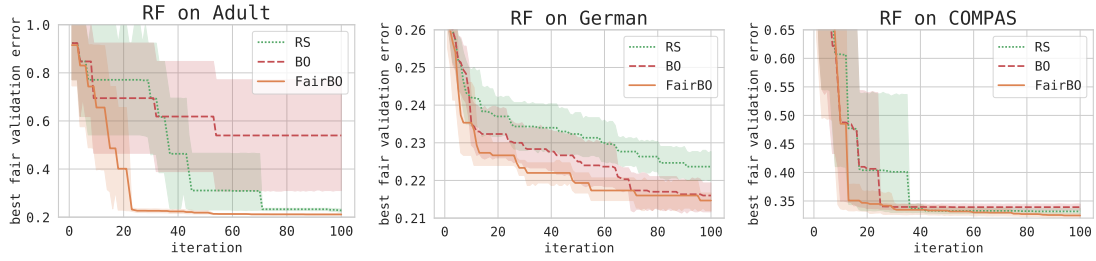


Figure 2: Comparison of RS, BO, and FairBO on the task of tuning RF on Adult, German, and COMPAS. The plot shows the validation error of the best feasible solution found as a function of the number of evaluations. The fairness constraint is $DSP \leq 0.05$. FairBO finds a more accurate fair model in fewer iterations.

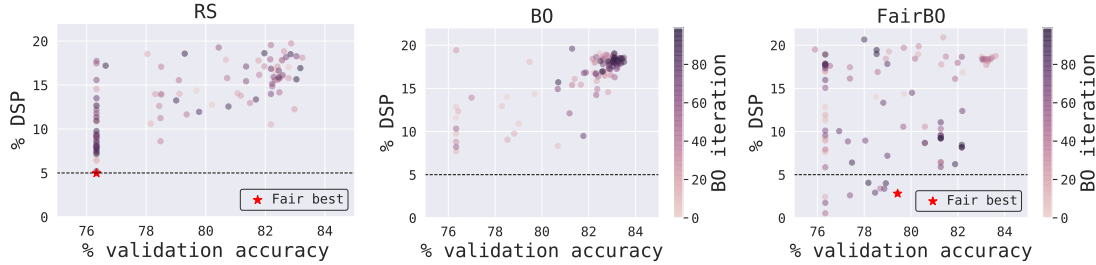


Figure 3: Comparison of RS, BO, and FairBO on the task of tuning RF on Adult. The horizontal line is the fairness constraint, set to $DSP \leq 0.05$, and darker dots correspond to later BO iterations. Standard BO can get stuck in high-performing yet unfair regions, failing to return a feasible solution (i.e., there is no fair best). RS is more robust than BO, but only finds a fair model with low validation accuracy.

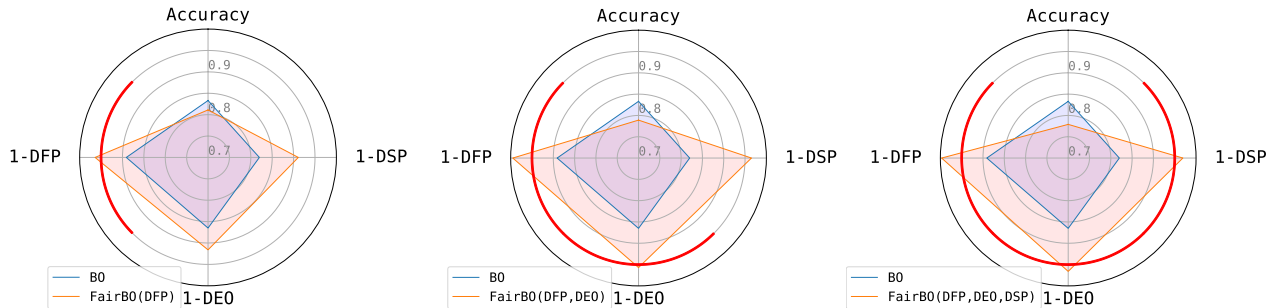


Figure 4: Best RF hyperparameter configuration found on Adult by BO and FairBO with respectively one, two, and three fairness constraints (left: $DFP \leq 0.05$; center: $\{DFP \leq 0.05, DEO \leq 0.05\}$; right: $\{DFP \leq 0.05, DEO \leq 0.05, DSP \leq 0.05\}$). The red arches represent the enforced level of fairness, i.e. the minimum level of fairness we can accept in order to satisfy the constraints. Compared to BO, FairBO can trade off accuracy for a fairer solution, and can do so with respect to all definitions simultaneously.

sensitive feature, we can bound the DSP as follows:

$$\begin{aligned}
 DSP(w) &= |P_X(w \cdot x \geq 0 | S = 0) - P_X(w \cdot x \geq 0 | S = 1)| \\
 &\leq P_X(-w_S \leq w \cdot x < 0 | S = 0).
 \end{aligned}$$

The idea is that, given x in subgroup $S = 0$ and being x' the same as x where we flipped the value of the sensitive feature to $S = 1$, we have $w \cdot x - w \cdot x' = -w_S$. Consequently, a smaller w_S helps obtain a less biased model. Indeed, unfairness correlates with the weight

assigned to the sensitive feature (or to the sum of the weights assigned to all the features correlated with the sensitive one), and regularization tends to alleviate this. Increasing the regularization in a cross-entropy loss will have a similar effect, giving progressively less data dependent models and steering DSP closer to zero. Similar insights apply to a more general setting of a non-linear parametric model (e.g., NN) and non-sensitive features: as if the hyperparameter $\alpha \rightarrow \infty \implies w \rightarrow 0$, which in turn means that

the model prediction approaches 0 regardless of the input features, satisfying all the fairness definitions in Section 2 in the process.

4.4 Model-agnostic and model-specific techniques

In the context of algorithmic fairness several *ad-hoc* methods have been proposed. We compare to the method of Zafar [60], Adversarial debiasing [63], and Fair Empirical Risk Minimization (FERM) [20].⁶ These methods enforce fairness during training and optimize the parameters of a linear model to make it both accurate and fair with respect to a fixed fairness definition. These methods are not model-agnostic and only apply to linear models. As alternative model-agnostic approaches we compare to SMOTE, which preprocesses the data by removing the sensitive feature and rebalancing the observations, as well as FERM preprocessing [20], which learns a fair representation of the data before fitting a linear model. We allocate 100 hyperparameter tuning iterations for all approaches.

Table 2 shows the best performing fair model found by FairBO on LL compared to the best performing fair model found by each baseline. As expected, FERM achieves higher accuracy, due to the constraint applied directly while training the parameters (as opposed to the hyperparameters) of the linear model. However, the gap in performance with FairBO is modest, and FairBO outperforms both Zafar and Adversarial Debiasing. While conceptually simple, FairBO emerges as a surprisingly competitive baseline that can outperform or compete against these highly specialized techniques. We note that all model-specific techniques tend to find solutions that are fairer than the required constraint. FairBO is also the best model-agnostic method, outperforming both SMOTE and FERM preprocessing. This shows that we can remove bias with a smaller impact on accuracy.

As FairBO only acts on the hyperparameters, it can be used *on top* of model-specific techniques to tune their own hyperparameters. Blindly tuning these hyperparameters can negatively impact the fairness of the resulting solution. We demonstrate this by combining FairBO with Zafar and Adversarial Debiasing, which we found to be sensitive to their hyperparameter settings (unlike FERM). Figure 7 shows that hyperparameter tuning on top of model-specific techniques yields better-performing fair solutions, and FairBO tends to find them more quickly than random search and standard BO. FairBO can be thus used to tune alternative fairness techniques, finding superior fair solutions.

4.5 Comparison with Multi-objective BO

Another natural methodology to find a model that is both fair and accurate is to use multi-objective (MO) optimization techniques (e.g., see parallel work in [18]). These methods attempt to find the Pareto front corresponding to different trade-offs between the objectives. Hence, they do not require the user to specify the fairness threshold(s) *a priori*. However, as they are trying to find the ensemble of solutions on the Pareto front (as opposed to the single best satisfying the constraint) they incur a much higher computational cost. Depending on the task at hand, it is not rare to have specific

regulations and laws imposing a well defined fairness metric and which range of values of the metrics is considered as acceptable – for example the 80% rule for Disparate Impact (see [24] for more details). In these situations, FairBO is able to provide a feasible solution faster than MO optimization techniques due to a more focused exploration of the feasible area of the domain.

We compare FairBO with two state-of-the-art MO Bayesian optimization (MOBO): (i) ParEGO [42], and (ii) a scalarization based method by Paria, et al., [46] implemented in the Dragonfly library [39]. MOBO aims to find the Pareto front of expensive multi-output opaque functions. Figure 8 shows the RF models found by MOBO on Adult with the same experimental setup and number of evaluations as Figure 3. MOBO methods are outperformed by FairBO in the feasible area ($DSP \leq 0.05$), because the *a priori* knowledge of the constraint enables FairBO to focus its attention on the (possibly small) feasible part of the Pareto front. In addition, FairBO samples the relevant fairness-performance space much more effectively than the multi-objective approaches. Finally, one should note that ParEGO and Dragonfly are designed to fully explore the Pareto front, and thus inevitably scale worse with respect to the number of objectives.

5 CONCLUSIONS

We showed that tuning model hyperparameters is surprisingly effective to mitigate unfairness in ML and proposed FairBO, a constrained Bayesian optimization framework to jointly tune ML models for accuracy and fairness. FairBO constitutes a strong baseline and is flexible. Indeed, it is model agnostic, can be used with arbitrary fairness definitions, and allows for multiple fairness definitions to be applied simultaneously. The proposed methodology empirically finds more accurate and fair solutions than data-debiasing techniques, while being competitive with the state-of-the-art in algorithm-specific fairness. We also showed that FairBO is preferable over standard BO when tuning the hyperparameters of specialized techniques. Finally, we demonstrated the importance of regularization hyperparameters in yielding fair and accurate models. Potential directions for future work include applying our framework to regression, image recognition, and natural language processing problems, and covering other fairness definitions, such as with continuous sensitive attributes.

REFERENCES

- [1] GPyOpt: A Bayesian optimization framework in Python. <http://github.com/SheffieldML/GPyOpt>, 2016.
- [2] A. Agarwal, A. Beygelzimer, M. Dudik, J. Langford, and H. Wallach. A reductions approach to fair classification. *ICML*, 2018.
- [3] A. Agarwal, M. Dudik, and Z. S. Wu. Fair regression: Quantitative definitions and reduction-based algorithms. *ICML*, 2019.
- [4] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias. *ProPublica*, 2016.
- [5] S. Barocas, M. Hardt, and A. Narayanan. Fairness and machine learning. *URL: www.fairmlbook.org*, 2018.
- [6] T. Bolukbasi, K. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. *NeurIPS*, 2016.
- [7] E. Brochu, V. M. Cora, and N. De Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. Technical report, preprint arXiv:1012.2599, 2010.
- [8] J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. *FAT**, 2018.
- [9] T. Calders, F. Kamiran, and M. Pechenizkiy. Building classifiers with independency constraints. *IEEE ICDM*, 2009.

⁶Code for Zafar from <https://github.com/mbilalzafar/fair-classification>; code for Adversarial Debiasing from <https://github.com/IBM/AIF360>; code for FERM from https://github.com/jmikko/fair_ERM.

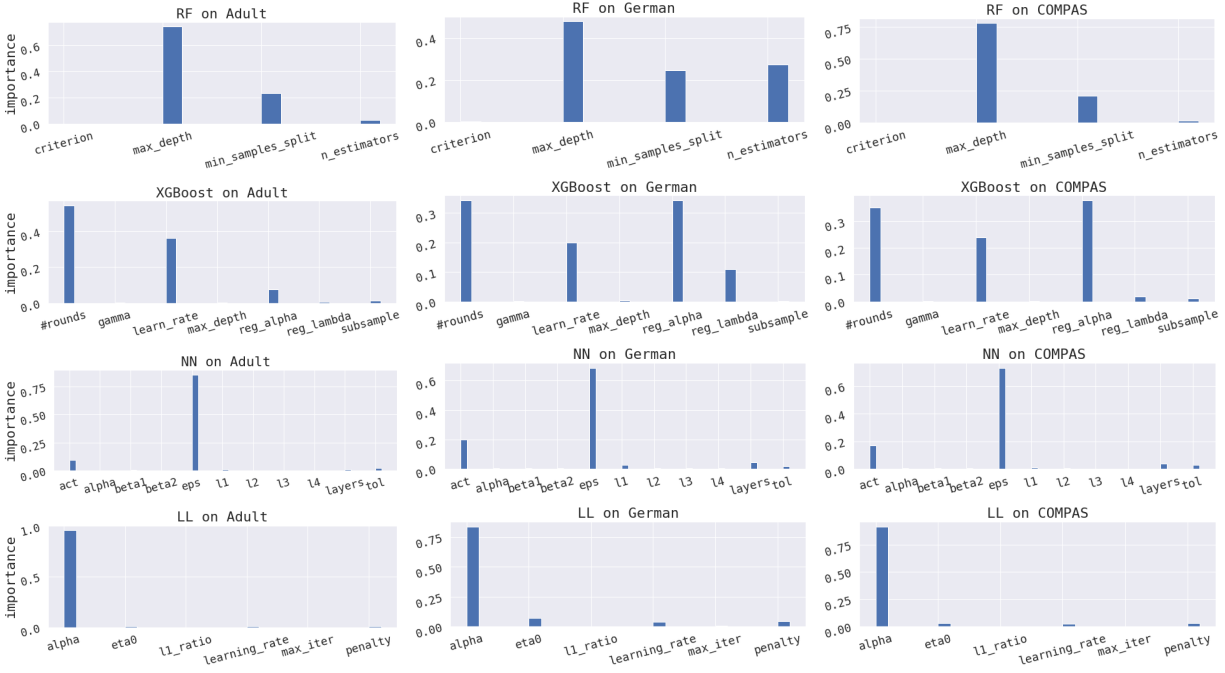


Figure 5: Hyperparameter importance on fairness: the role of each tuned hyperparameter on model unfairness (DSP) is evaluated with fANOVA [34]. Statistics are collected from 100 iterations of random search and 10 seeds. Regularization hyperparameters tend to impact fairness the most (e.g., the most relevant hyperparameter in LL is precisely the regularization factor alpha).

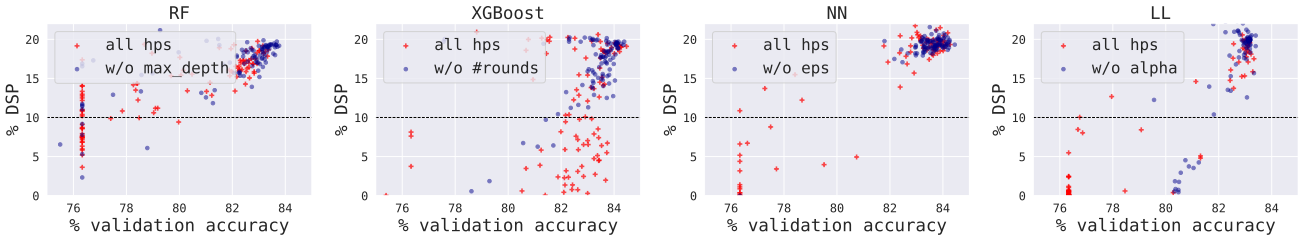


Figure 6: Unfairness vs Accuracy on Adult for 100 random hyperparameter configurations of RF, XGBoost, NN, and LL. For each algorithm, we either vary all hyperparameters or fix the one most correlated with fairness (detected by fANOVA [34]). As expected, fixing these hyperparameters limits the ability of FairBO to provide fair and accurate solutions, yielding fewer fair configurations (fewer blue dots fall in the feasible area below the horizontal line).

Table 2: Validation error of the best fair models for model-specific (first three rows) and model-agnostic fairness methods. We use the fairness constraint, $DSP \leq 0.1$. The best method from each category is highlighted in bold.

Method	Adult	German	COMPAS
FERM	0.164 ± 0.010	0.185 ± 0.012	0.285 ± 0.009
Zafar	0.187 ± 0.001	0.272 ± 0.004	0.411 ± 0.063
Adversarial	0.237 ± 0.001	0.227 ± 0.008	0.327 ± 0.002
FERM preprocess	0.228 ± 0.013	0.231 ± 0.015	0.343 ± 0.002
SMOTE	0.178 ± 0.005	0.206 ± 0.004	0.321 ± 0.002
FairBO (ours)	0.175 ± 0.007	0.196 ± 0.005	0.307 ± 0.001

[10] A. Caliskan, J. J. Bryson, and A. Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 2017.

[11] F. Calmon, D. Wei, B. Vinzamuri, K. N. Ramamurthy, and K. R. Varshney. Optimized pre-processing for discrimination prevention. *NeurIPS*, 2017.

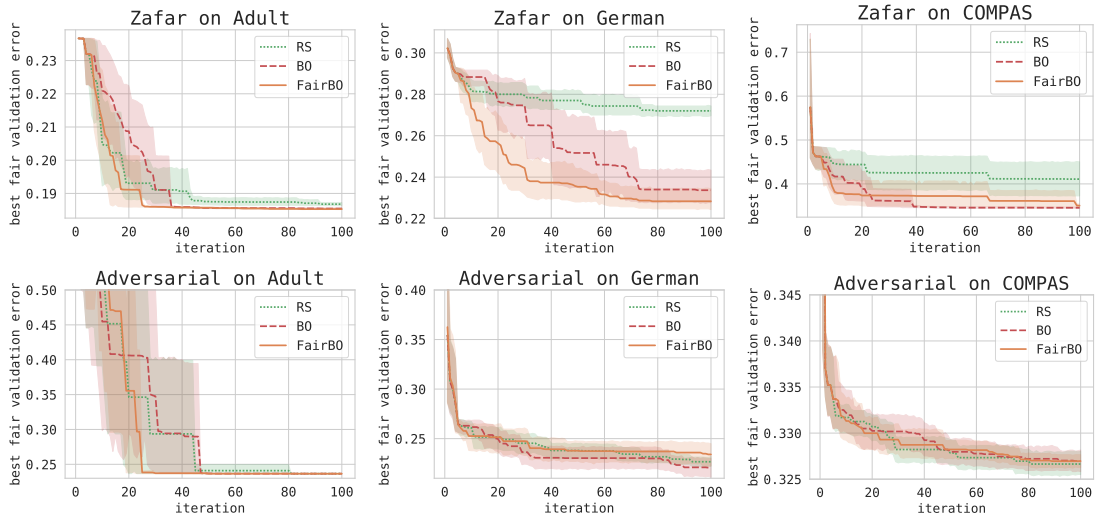


Figure 7: Comparison of RS, BO, and FairBO where Zafar and Adversarial Debiasing are used as base learners. We use the fairness constraint, $DSP \leq 0.1$. Hyperparameter tuning on top of model-specific techniques finds better performing fair solutions. The use of FairBO tends to do so in fewer iterations (horizontal axis) than when using RS or BO.

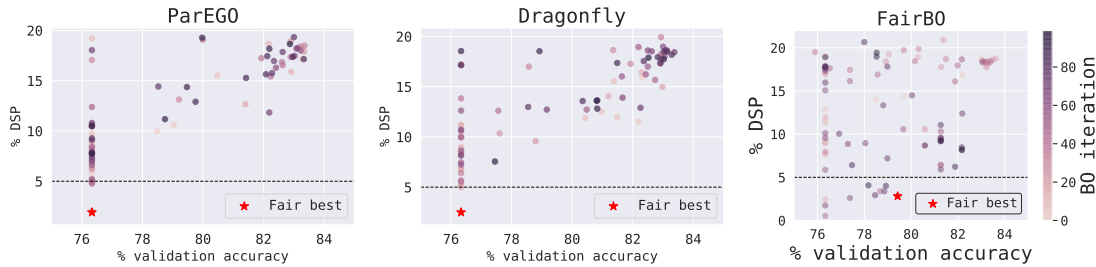


Figure 8: ParEGO and Dragonfly tuning RF on Adult compared to FairBO. The horizontal line is the fairness constraint ($DSP \leq 0.05$), and darker dots are later BO iterations. MOBO methods find a better Pareto Front but fail to focus on the feasible region.

[12] L. E. Celis, D. Straszak, and N. K. Vishnoi. Ranking with fairness constraints. *ICALP*, 2018.

[13] J. Chakraborty, S. Majumder, Z. Wu, and T. Menzies. Fairway: SE principles for building fairer software. *arXiv:2003.10354*, 2020.

[14] J. Chakraborty, T. Xia, F. M. Fahid, and T. Menzies. Software engineering for fairness: A case study with hyperparameter optimization. *arXiv:1905.05786*, 2019.

[15] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 2002.

[16] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. *ACM SIGKDD*, 2016.

[17] S. Chiappa and W. S. Isaac. A causal bayesian networks viewpoint on fairness. *arXiv:1907.06430*, 2019.

[18] A. F. Cruz, P. Saleiro, C. Belém, C. Soares, and P. Bizarro. A bandit-based algorithm for fairness-aware hyperparameter optimization. *arXiv:2010.03665*, 2020.

[19] C. Dimitrakakis, Y. Liu, C. D. Parkes, and G. Radanovic. Bayesian Fairness. *AAAI*, pages 509–516, 2019.

[20] M. Donini, L. Oneto, S. Ben-David, J. S. Shawe-Taylor, and M. Pontil. Empirical risk minimization under fairness constraints. *NeurIPS*, 2018.

[21] D. Dua and C. Graff. UCI machine learning repository, 2017.

[22] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. *ITCS*, 2012.

[23] N. Erickson, J. Mueller, A. Shirkov, H. Zhang, P. Larroy, M. Li, and A. Smola. Autogluon-tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505*, 2020.

[24] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian. Certifying and removing disparate impact. *ACM SIGKDD*, 2015.

[25] J. R. Foulds, R. Islam, K. N. Keya, and S. Pan. Bayesian Modeling of Intersectional Fairness: The Variance of Bias. 2020.

[26] S. A. Friedler, C. Scheidegger, and S. Venkatasubramanian. On the (im)possibility of fairness. *arXiv preprint arXiv:1609.07236*, 2016.

[27] S. A. Friedler, C. Scheidegger, S. Venkatasubramanian, S. Choudhary, E. P. Hamilton, and D. Roth. A comparative study of fairness-enhancing interventions in machine learning. *FAT**, 2019.

[28] J. Gardner, M. Kusner, Z. Xu, K. Weinberger, and J. Cunningham. Bayesian optimization with inequality constraints. *ICML*, 2014.

[29] M. A. Gelbart, J. Snoek, and R. P. Adams. Bayesian optimization with unknown constraints. *UAI*, 2014.

[30] D. Golovin, B. Solnik, S. Moitra, G. Kochanski, J. E. Karro, and D. Sculley. Google Vizier: A service for black-box optimization. *KDD*, 2017.

[31] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. *NeurIPS*, 2016.

[32] P. Hennig and C. J. Schuler. Entropy search for information-efficient global optimization. *JMLR*, 2012.

[33] J. M. Hernández-Lobato, M. A. Gelbart, M. W. Hoffman, R. P. Adams, and Z. Ghahramani. Predictive entropy search for Bayesian optimization with unknown constraints. *ICML*, 2015.

[34] F. Hutter, H. Hoos, and K. Leyton-Brown. An efficient approach for assessing hyperparameter importance. *ICML*, 2014.

[35] S. Jabbari, M. Joseph, M. Kearns, J. Morgenstern, and A. Roth. Fairness in reinforcement learning. *ICML*, 2017.

[36] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 1998.

- [37] F. Kamiran and T. Calders. Data preprocessing techniques for classification without discrimination. *KAIS*, 2012.
- [38] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma. Fairness-aware classifier with prejudice remover regularizer. *ECML PKDD*, 2012.
- [39] K. Kandasamy, K. R. Vysyaraju, W. Neiswanger, B. Paria, C. R. Collins, J. Schneider, B. Póczos, and E. P. Xing. Tuning hyperparameters without grad students: Scalable and robust bayesian optimisation with dragonfly. *Journal of Machine Learning Research*, 21(81):1–27, 2020.
- [40] M. Kearns, S. Neel, and Z. S. Roth, A. and Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. 2018.
- [41] J. Kleinberg. Inherent trade-offs in algorithmic fairness. *SIGMETRICS*, 2018.
- [42] J. Knowles. Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.
- [43] M. MacCarthy. Standards of fairness for disparate impact assessment of big data algorithms. *Cumb. L. Rev.*, 48:67, 2017.
- [44] J. Mockus, V. Tiesis, and A. Zilinskas. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 1978.
- [45] L. Oneto, M. Donini, M. Pontil, and J. Shawe-Taylor. Randomized learning and generalization of fair and private classifiers: From PAC-Bayes to stability and differential privacy. *Neurocomputing*, 2020.
- [46] B. Paria, K. Kandasamy, and B. Póczos. A flexible framework for multi-objective bayesian optimization using random scalarizations. In *UAI*, 2019.
- [47] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *JMLR*, 2011.
- [48] V. Perrone, I. Shcherbatyi, R. Jenatton, C. Archambeau, and M. Seeger. Constrained Bayesian optimization with max-value entropy search. *NeurIPS Meta Learning Workshop*, 2019.
- [49] V. Perrone, H. Shen, A. Zolic, I. Shcherbatyi, A. Ahmed, T. Bansal, M. Donini, F. Winkelmolen, R. Jenatton, J. B. Faddoul, B. Pogorzelska, M. Miladinovic, K. Ken-thapadi, M. Seeger, and C. Archambeau. Amazon sagemaker automatic model tuning: Scalable black-box optimization. Technical report, arXiv:2012.08489, 2020.
- [50] F. Pfisterer, S. Coors, J. Thomas, and B. Bischl. Multi-objective automatic machine learning with autoxgboostmc. *arXiv:1908.10796*, 2019.
- [51] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- [52] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *IEEE*, 2016.
- [53] J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M. Patwary, M. Prabhat, and R. Adams. Scalable Bayesian optimization using deep neural networks. *ICML*, 2015.
- [54] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58:3250–3265, 2012.
- [55] P. S. Thomas, B. C. da Silva, A. G. Barto, S. Giguere, Y. Brun, and E. Brunskill. Preventing undesirable behavior of intelligent machines. *Science*, 366(6468):999–1004, 2019.
- [56] S. Verma and J. Rubin. Fairness definitions explained. *FairWare*, 2018.
- [57] Z. Wang and S. Jegelka. Max-value entropy search for efficient Bayesian optimization. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 3627–3635, 2017.
- [58] B. Woodworth, S. Gunasekar, M. I. Ohannessian, and N. Srebro. Learning non-discriminatory predictors. *COLT*, 2017.
- [59] M. Yaghini, H. Heidari, and A. Krause. A human-in-the-loop framework to construct context-dependent mathematical formulations of fairness. *arXiv preprint arXiv:1911.03020*, 2019.
- [60] M. B. Zafar, I. Valera, M. Gomez-Rodriguez, and K. P. Gummadi. Fairness constraints: Mechanisms for fair classification. *AISTATS*, 2017.
- [61] M. B. Zafar, I. Valera, M. Gomez-Rodriguez, and K. P. Gummadi. Fairness constraints: A flexible approach for fair classification. *JMLR*, 2019.
- [62] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork. Learning fair representations. *ICML*, 2013.
- [63] B. H. Zhang, B. Lemoine, and M. Mitchell. Mitigating unwanted biases with adversarial learning. *AIES*, 2018.

A OPTIMIZED HYPERPARAMETERS

A.1 Algorithms

We considered the problem of tuning four popular ML algorithms, as implemented in `scikit-learn`: XGBoost (XGB), Random Forest (RF), Neural Network (NN), Linear Learner (LL). In this section, we give more details on the search space over which each hyperparameter was optimized.

XGBoost. We consider a 7-dimensional search space: number of boosting rounds in $\{1, 2, \dots, 256\}$ (log scaled), learning rate in $[0.01, 1.0]$ (log scaled), minimum loss reduction to partition leaf node γ in $[0.0, 0.1]$, L1 weight regularization α in $[10^{-3}, 10^3]$ (log scaled), L2 weight regularization λ in $[10^{-3}, 10^3]$ (log scaled), subsampling rate in $[0.01, 1.0]$, maximum tree depth in $\{1, 2, \dots, 16\}$.

RF. We consider a 4-dimensional search space: number of trees in $\{1, 2, \dots, 64\}$ (log scaled), tree split threshold in $[0.01, 0.5]$ (log scaled), tree maximum depth in $\{1, 2, 3, 4, 5\}$, criterion for quality of split in {Gini, Entropy}.

NN. We consider an 11-dimensional search space: number of layers in $\{1, 2, 3, 4\}$, each layer size in $\{2, 3, \dots, 32\}$ (log scaled), activation in {Logistic, Tanh, ReLU}, tolerance in $[10^{-5}, 10^{-2}]$ (log scaled), L2 regularization in $[10^{-6}, 0.1]$ (log scaled), and Adam parameters: initial learning rate ϵ in $[10^{-6}, 10^{-2}]$ (log scaled), β_1 and β_2 in $[10^{-3}, 0.99]$ (log scaled).

LL. We consider a 6-dimensional search space: iteration count in $\{1, 2, \dots, 128\}$, regularization type in {L1, L2, ElasticNet}, Elastic Net mixing parameter in $[0, 1]$, regularization factor α in $[10^{-3}, 10^3]$ (log scaled), initial learning rate ϵ in $[10^{-4}, 0.1]$ (log scaled), learning rate schedule in {Constant, Optimal, Invscaling, Adaptive}.

A.2 Baselines

In addition to the hyperparameters of the tuned algorithms, we allocated 100 iterations of random search to tune each baseline, namely FERM, Zafar, Adversarial, FERM preprocess and SMOTE.

FERM. We considered 2 hyperparameters for FERM. The L2 regularization coefficient C in $[0.001, 10.0]$ (log scaled), and FERM’s epsilon-fairness threshold in $[0.0, 1.0]$.

Zafar. We tuned 3 hyperparameters: L1 regularization coefficient in $[10^{-3}, 10.0]$ (log scaled), L2 regularization coefficient in $[0.001, 10.0]$ (log scaled), and Zafar’s epsilon-fairness threshold in $[0.0, 1.0]$.

Adversarial. We tuned 4 different hyperparameters: the adversary loss weight in $[0.0, 1.0]$, the number of epochs for training in $\{1, 2, \dots, 500\}$ (log scaled), the batch size in $\{32, 33, \dots, 256\}$ (log scaled), and the number of hidden units of the network between in $\{20, 21, \dots, 5000\}$ (log scaled).

FERM preprocess. FERM preprocessing learns a fair representation of the dataset, which is then fed to LL. Hence, we tuned the same 6 hyperparameters as per the original LL.

SMOTE. In addition to the 6 hyperparameters of LL, we jointly tuned 2 hyperparameters controlling the degree of dataset rebalancing: oversampling rate of the less frequent class in $[0.3, 1.0]$ and number of neighbors to generate synthetic examples in $\{1, 2, \dots, 20\}$.

B ADDITIONAL EXPERIMENTS

Compared to Random Search (RS) and standard BO, FairBO explores the fair regions of the hyperparameter space more quickly and tends to return a more accurate, fair solution. This appendix presents additional experiments with different fairness thresholds, definitions and algorithms. We also study the impact of each hyperparameter on the deviation from equal opportunity (DEO) of the resulting model.

Fairness thresholds. We investigate the impact of varying the fairness threshold ϵ . We repeat the experiment of tuning a RF model on Adult, German and COMPAS, this time with a looser fairness constraint of $\text{DSP} \leq 0.15$. Figure 9 shows the validation error of the fair solution on the three datasets. As expected, the performance gap between FairBO and the baselines is still clear but overall less pronounced compared to the experiments with a stricter fairness constraint $\text{DSP} \leq 0.05$. Additionally, due to the looser constraint, the accuracy of the best fair solution is significantly higher in Adult and COMPAS. At the same time, FairBO still tends to find a well-performing fair model more quickly than RS and BO. Figure 10 illustrates the behavior on Adult, indicating that standard BO can still get stuck in high-performing but unfair regions. Although BO also finds a fair solution, this is less accurate than the one found by FairBO. RS also tends to require more resources to find an accurate and fair solution.

Fairness definitions. FairBO can be applied directly to arbitrary fairness definitions. In this section, we repeat the RF experiments by replacing the initial, strict constraint on statistical parity ($\text{DSP} \leq 0.05$) with an analogous constraint on equal opportunity ($\text{DEO} \leq 0.05$). As previously, 100 iterations are allocated to RS, BO and FairBO. Figure 11 shows the best validation error of the fair solution found after each hyperparameter evaluation on the three datasets. Consistently with the DSP constraint, FairBO finds fair and accurate solutions more quickly than BO and RS. In addition, constraining DEO on COMPAS allows for fair solutions with higher accuracy compared to constraining DSP. This is not surprising as DEO is generally a better proxy of accuracy than DSP (a perfect classifier has DEO equal to zero, unlike DSP).

FairBO can also handle multiple fairness definitions simultaneously. As in the experiments with RF in the main paper, we consider 100 iterations of standard BO and FairBO on the problem of tuning NN, XGBoost and LL on Adult while satisfying the three fairness definitions. Specifically, we impose that DFP, DEO and DSP should all be less than 0.05. Results are averaged over 10 independent repetitions. Figure 12 shows accuracy and three fairness metrics of the returned fair solutions, with the red arches indicating the constraints. FairBO allows us to trade off a relatively small degree of accuracy to get a more fair solution. Interestingly, when FairBO is applied to XGBoost, the fair solution comes with the slightest accuracy loss.

Tuning XGboost and NN. Figure 13 compares RS with BO and FairBO on the problem of tuning XGboost and a NN on Adult, COMPAS, and German. Solutions are considered fair if they satisfy the strict requirement of $\text{DSP} \leq 0.05$. As in the case of RF, FairBO tends to find an accurate and fair solution faster than baselines. Consistently with the multiple constraint experiment, tuning XGBoost allows for more accurate fair solutions than RF and NN.

Acquisition functions. Being based on a general constrained BO framework, FairBO can be implemented through different acquisition functions beyond cEI. An example is cMES, a constrained extension of the max-value entropy search [47]. Figure 14 demonstrates this by comparing FairBO implemented through either cEI and cMES on the problem of tuning RF on Adult, COMPAS, and German. Solutions are considered fair if they satisfy the requirement of $\text{DSP} \leq 0.15$. The results confirm that cMES is another viable implementation of FairBO which achieves consistent performance on these benchmarks. A systematic comparison of the two acquisition functions is provided in [47].

Hyperparameters and equal opportunity. Previous experiments shed light on the role of hyperparameter tuning on unfairness. We now study the contribution of each hyperparameter when unfairness is defined as difference in equal opportunity (DEO). As in the DSP experiments in the main paper, for each algorithm we study hyperparameter importance via fANOVA. Hyperparameter configurations and unfairness metrics are collected from 100 iterations of random search and 10 random seeds for each dataset. Figure 15 shows hyperparameter importance on DEO, confirming the results obtained with DSP and indicating that the hyperparameters controlling regularization tend to play the largest role. In the case of RF, the most important hyperparameter is the maximum tree depth on Adult and COMPAS, and number of trees on German; for XGBoost, the L1 weight regularizer α , number of boosting rounds or learning rate are the most important; for NN, the most impactful is Adam’s initial learning rate eps (as the number of epochs is kept fixed to the default value); finally, the most relevant hyperparameter in LL is, once again, exactly the regularization factor α .

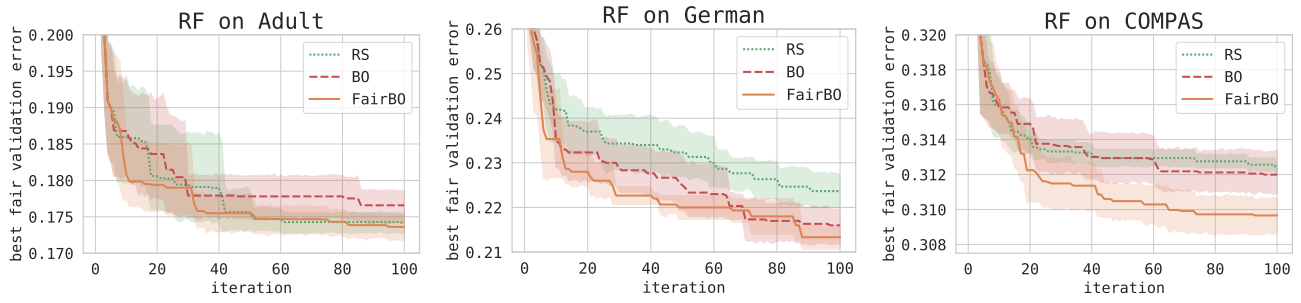


Figure 9: Comparison of RS, BO, and FairBO over the validation error (vertical axis) of the best feasible solution on Adult, German, and COMPAS data using RF as base model. The fairness constraint is set to a looser value of $DSP \leq 0.15$. FairBO tends to find a well-performing fair model in fewer iterations than RS and BO.

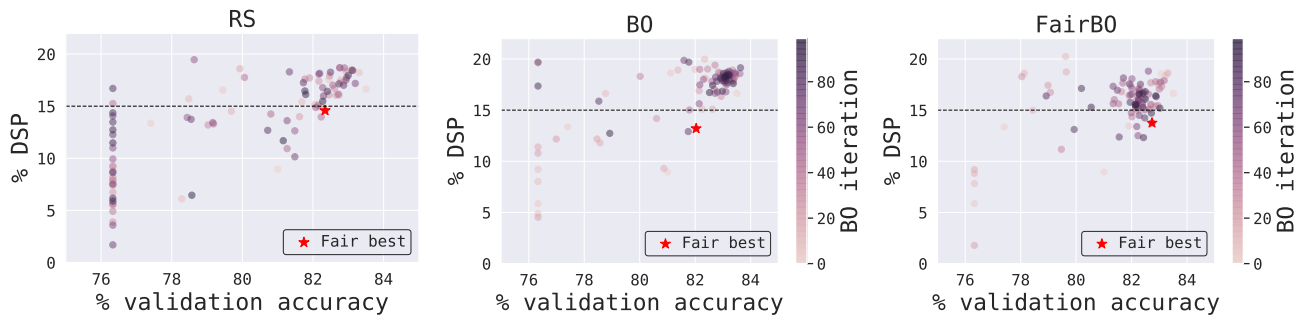


Figure 10: Comparison of FairBO, BO, and RS when tuning RF on Adult. The horizontal line is the fairness constraint, set to a looser value of $DSP \leq 0.15$. Darker dots correspond to later BO iterations. Both RS and BO find a less accurate fair solution after 100 iterations. Additionally, BO can get stuck in high-performing yet unfair regions.

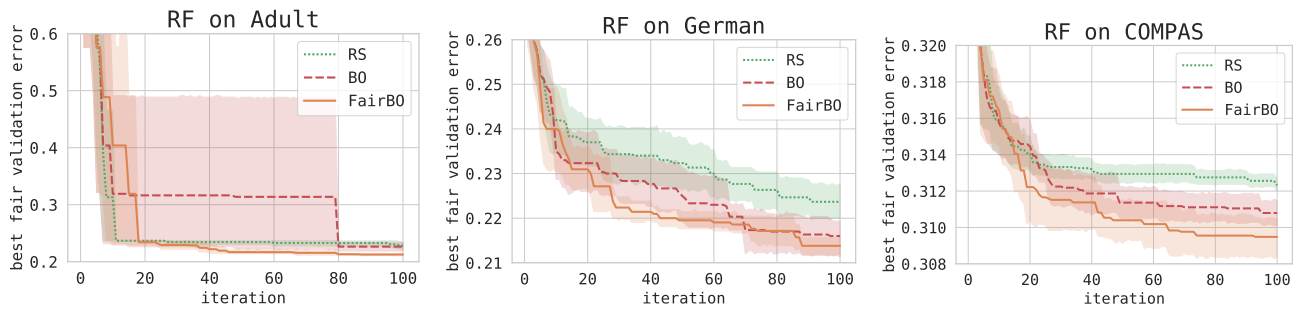


Figure 11: Comparison of RS, BO, and FairBO over the validation error (vertical axis) of the best feasible solution on Adult, German, and COMPAS data using RF as base model. The fairness constraint is set to $DEO \leq 0.05$. FairBO tends to find a well-performing fair model more with fewer iterations than RS and BO.

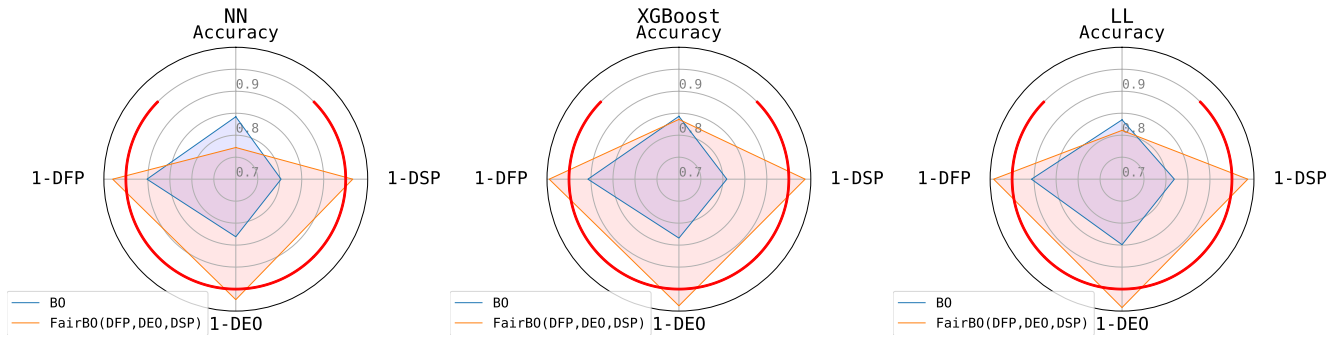


Figure 12: Best hyperparameter configuration found on Adult by BO and FairBO on NN, XGBoost, and LL with three simultaneous fairness constraints, represented by the red arches: $\{DFP < 0.05, DEO < 0.05, DSP < 0.05\}$ (i.e., the minimum level of fairness we can accept). Compared to BO, FairBO can trade off accuracy for a fairer solution with respect to each definition. When applied to XGBoost, a slight accuracy degradation is enough to find solutions that are fair across all definitions.

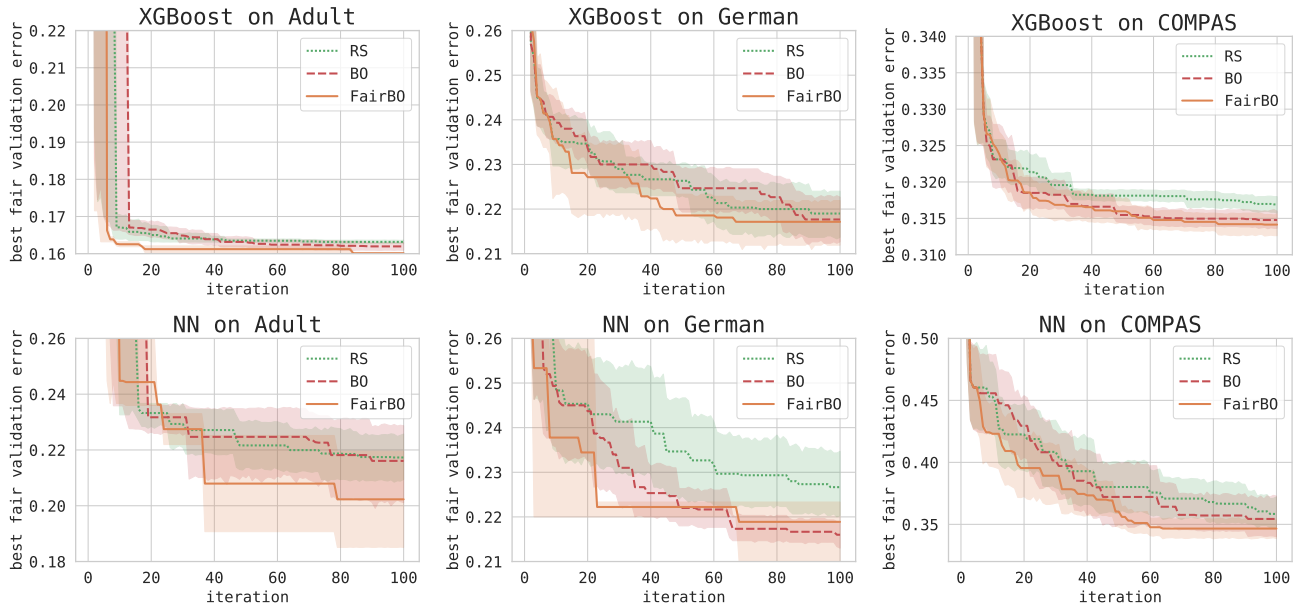


Figure 13: Comparison of RS, BO, and FairBO over the validation error (vertical axis) of the best feasible solution on Adult, German, and COMPAS data using XGBoost and NN as base models. The fairness constraint is set to $DSP \leq 0.05$. FairBO tends to find a well-performing fair model in fewer iterations than RS and BO.

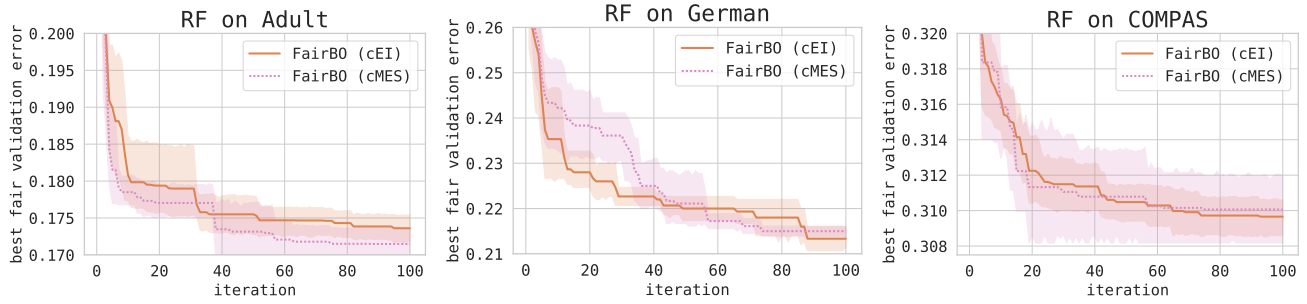


Figure 14: Comparison of cEI and cMES, two instantiations of FairBO based on the constrained expected improvement and constrained max-value entropy search acquisition function, respectively. We compare the validation error (vertical axis) of the best feasible solution on Adult, German, and COMPAS with a constraint set to $DSP \leq 0.15$. The two instantiations of FairBO achieve comparable performance.

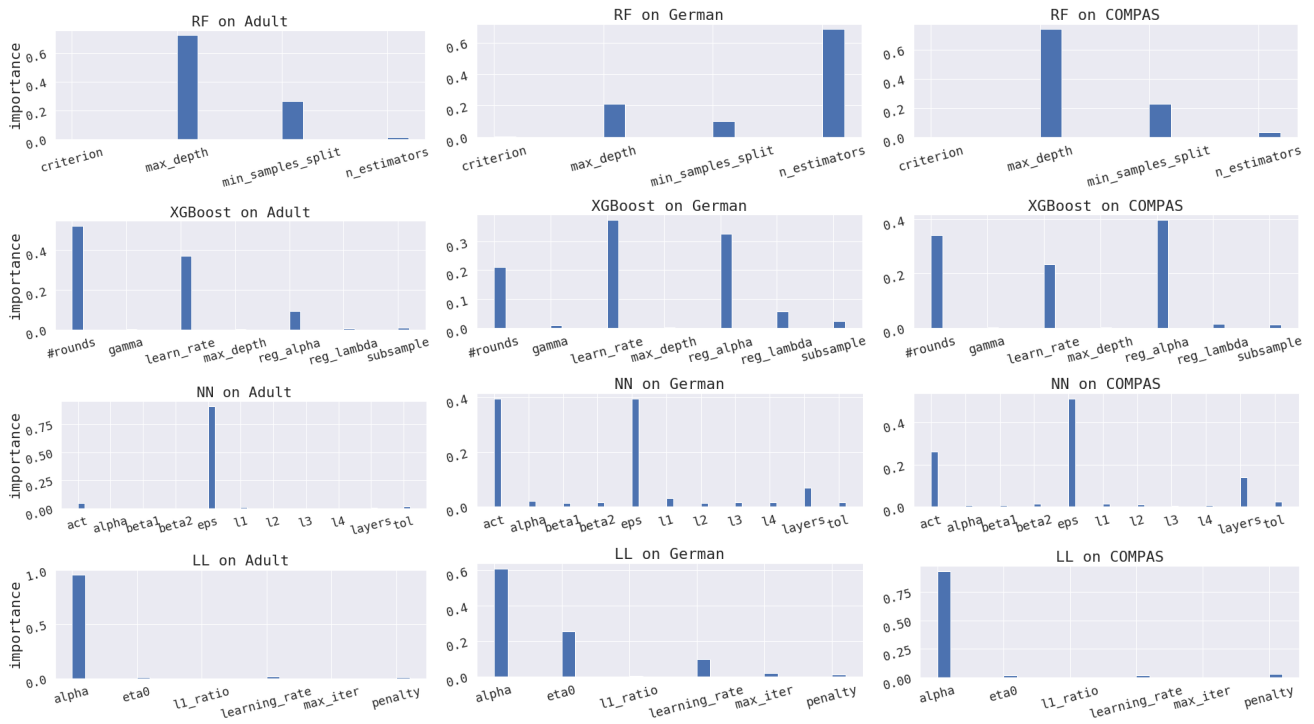


Figure 15: Hyperparameter importance on equal opportunity. The role of each hyperparameter on the unfairness (DEO) of the resulting model is evaluated via fANOVA. Statistics are collected from 100 iterations of random search and 10 seeds. Regularization hyperparameters tend to impact fairness the most (e.g., LL’s most important hyperparameter is precisely the regularization factor alpha).