Universiteit Leiden

Master Computer Science

Predicting illegal ship breaking via fairness-aware classifier optimisation

Name:	Louka Wijne
Student ID:	s2034697
Date:	03/09/2024
Specialisation:	Artificial Intelligence
1st supervisor:	António Pereira Barata
2nd supervisor:	Jan N. van Rijn

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Within the maritime oversight domain, illegal ship breaking is an unlawful type of ship disposal process that constitutes a serious threat to human life and environmental sustainability. To combat these hazardous events, the Dutch Human Environment and Transport Inspectorate aims to generate a reliable supervised classification model trained on historical data to predict which currently sailing ships near their end-of-life are the most probable candidates for such activities. However, given inherent systemic biases present in the data used for learning, the resulting predictive models tend to overemphasise certain ship country flags, leading to poor generalisation and loss of performance under concept drift. To address this, the current work explores the application of fair machine learning algorithms to the task, aiming to develop models that are as unbiased as possible while maintaining robust predictive performance. Specifically, this study investigates three traditional machine learning algorithms: (1) Random Forest, (2) Logistic Regression, and (3) Neural Network, and three fair machine learning algorithms: (1) Fair Random Forest, (2) Fair Logistic Regression, and (3) Fair Adversarial Learning. These algorithms were chosen for their ability to balance predictive performance with fairness by incorporating performance-fairness trade-off hyperparameters. Comprehensive experimentation was conducted to assess the feasibility of our task. We experimented with traditional and fair machine learning algorithms optimised for classification performance and traditional and fair machine learning algorithms optimised for both classification performance and fairness simultaneously, using fairness-aware hyperparameter optimisation with a functional model-agnostic fairness-aware objective function. Furthermore, benchmark experiments were performed to validate our experimental design. Our results show that Fair Random Forest offers a reliable solution for balancing performance and fairness, achieving the broadest range of predictive performance and fairness scores. For both traditional and fair machine learning algorithms, their fairness and tunability in regard to the performance-fairness trade-off was severely improved by enhancing these algorithms with fairness-aware hyperparameter optimisation, by incorporating a compound performance-fairness objective function into the hyperparameter optimisation step. Ultimately, this study not only advances the understanding of the trade-off between predictive performance and fairness in this domain-specific machine learning problem but also demonstrates the broader applicability of pairing fair machine learning algorithms with fairness-aware hyperparameter optimisation to enhance control over this critical balance.

Contents

1 Introduction

Machine learning models have become increasingly ubiquitous in aiding human decision-making over the years (Lu and Yin, 2021). These models are trained on data, which means they can carry over any inherent biases present in that data (Mehrabi et al., 2022). When not adequately addressed, this preservation of bias can have severe adverse implications(European Commision, 2024). Notorious examples include Amazon abandoning their automated recruitment system due to discrimination against women (Goodman, 2018), the Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) predicting black defendants to be at a higher risk of re-offending than white defendants with all other variables controlled (Larson et al., 2016), and a model used by US hospitals underestimating thee medical needs of black patients, thereby disqualifying them from extra care disproportionately with respect to white patients with the same medical needs (Grant, 2022). However, not all biased data translate to ethical concerns; consider the present use-case within the international maritime transport domain, which is the focus of our work.

Certain countries are known for poorly implementing international maritime law, which translates to financial gain for the companies involved as they spend less on compliance. Moreover, these countries offer popular low-cost "last-voyage" packages for ships nearing the end of their life. This often results in ships being illegally beached rather than properly dismantled, leading to severe environmental damage, hazardous working conditions, and health risks for workers (NGO Shipbreaking Platform, 2024). Consequently, the data describing this reality is biased towards these specific country flags (known as flags of convenience), under which the illegally-beached ships were sailing. In the Netherlands, the Human Environment and Transport Inspectorate (ILT) is the legal authority responsible for, among other duties, ensuring compliance with proper ship dismantling practices to mitigate the environmental and health hazards associated with illegal beaching (Ministerie van Infrastructuur en Milieu, 2023). The ILT has developed a model to predict which ships will be beached, but it is biased towards ships sailing under flags of convenience. The concern over this specific bias is non-ethical, but rather that it makes the model myopic, overly focusing on flags of risk, which may cause it to miss ships sailing under non-target flags and increase false positives by overselecting on target flags. Additionally, ships can change flags at the last moment to evade detection and accountability (de Bruin et al., 2022), meaning the model may not always work with accurate flag information. To address this critical issue, an unbiased model is essential.

Learning unbiased models from biased data is the goal of fairness algorithms (Weerts et al., 2023). These algorithms can be categorised into three distinct approaches based on when they are applied within the model learning process: (1) pre-processing, (2) in-processing, and (3) post-processing. Pre-processing approaches adjust the data before feeding it into a model to eliminate certain associations between sensitive attributes and targets. In-processing approaches incorporate fairness constraints within the algorithm by adapting existing algorithms or using wrapper methods. Post-processing approaches adjust trained models through post-processing predictions or by modifying model parameters. Pre-processing and post-processing methods for bias mitigation do not alter the machine learning model, preserving the use of existing li-

braries (Caton and Haas, 2024), but may reduce model interpretability. In contrast, in-processing methods optimise fairness during training, only requiring an optimisation function towards a fairness measure in conjunction with the traditional classification performance.

In this paper, we propose to address the critical issue of illegal ship breaking through fair classifier learning. We will do so by implementing three fairness approaches: (1) Fair Random Forest, a probabilistic tree learning algorithm that optimises both performance and fairness; (2) Fair Logistic Regression, a logistic regression model which minimises classification error subject to fairness constraints; and (3) Fair Adversarial Learning, a neural network which leverages adversarial training to classify while remaining unbiased. We chose these algorithms for their ability to balance fairness and predictive performance (i.e., performance-fairness trade-off), which is especially important for this use case. We formulate our problem statement as follows:

> *How can we best minimise the bias present in the illegal ship breaking prediction model of the ILT whilst maintaining adequate classification performance?*

To make the problem statement tractable, we decompose it into three research questions:

1. How do traditional machine learning algorithms perform in terms of both predictive performance and fairness?
2. How do the fairness parameters of fair machine learning algorithms influence their predictive performance and fairness?
3. How do the predictive performance, fairness, and the trade-off between these measures change for traditional and fair machine learning algorithms when employing fairness-aware hyperparameter optimisation?

By answering the aforementioned research questions and therein addressing our problem statement, we propose the following three contributions: (1) improving the fairness of the ILT's ship breaking model, (2) proposing a functional model-agnostic fairness-aware objective function, and (3) proposing the practice of employing fair machine learning algorithms in combination with fairness-aware hyperparameter optimisation.

The remainder of this paper is structured as follows: Section 2 defines our problem formally. Section 3 reviews related work from the fairness literature. Section 4 describes the methods that we employ. Section 5 describes the data used in this study. Section 6 outlines the methods and experimental setup employed to answer our research questions. Section 7 presents our results. Section 8 includes a discussion of our findings, addresses some limitations, and suggests directions for future research. Finally, Section 9 concludes the paper.

2 Problem Definition

We aim to develop a machine learning classifier that predicts whether a ship near the end of its life will be illegally dismantled. A key priority is ensuring fairness in these predictions by removing any undue influence of the flag under which a ship sails; specifically, whether it is a *flag of convenience* or an alternative flag. We must assess the performance and fairness of the model directly based on its outputs rather than relying on a predetermined decision threshold. By doing so, we can prevent any biases that can be brought about by artificial thresholds that would distort justice. Ultimately, our goal is to create a fair classifier that offers control over the trade-off between predictive performance and fairness. This adjustable balance is essential since no single solution may meet both requirements simultaneously, consequently enabling the appropriate stakeholders to define their trade-off point of interest.

Formally, we consider a dataset with n instances, m attributes, and two classes. We assume the existence of an additional sensitive attribute. Accordingly, instances are represented by (x_i, y_i, s_i), for $i = 1, 2, \ldots, n$, in which $x \in X \subseteq \mathbb{R}^m$, $y \in \{y_+, y_-\} \subseteq Y$, and $s \in \{s_+, s_-\} \subseteq S$. In practice, this amounts to a matrix of size $n \times m$ (i.e., X), and a pair of n-length column vectors (i.e., Y and S). Furthermore, we define a classification model as a mapping function $f \in F : x \in X \subseteq \mathbb{R}^m \rightarrow z \in Z \subseteq \mathbb{R} \xrightarrow{t \in T \subseteq \mathbb{R}} \hat{y} \in \{y_+, y_-\} \subseteq \hat{Y}$, where prediction scores Z may be reduced — via a decision threshold t — to label predictions $\hat{Y}$, such that $z \rightarrow y_+$ if $z > t$, and $z \rightarrow y_-$ otherwise. Lastly, function f must be *learned* from the dataset via a machine learning algorithm $h \in H$ of *hyperparameters* $\lambda \in \Lambda$, formally $h_\lambda : (X, Y, S) \rightarrow f_{h_\lambda}$. Note: while learning algorithm h takes S as part of its input, classification model f does not. Our goal is, therefore, to find a *fair* classification model f of which the output Z must simultaneously (1) maximise $P[(Z|Y{=}y_+) > (Z|Y{=}y_-)]$ and (2) minimise $|P[(Z|S{=}s_+) > (Z|S{=}s_-)] - P[(Z|S{=}s_-) > (Z|S{=}s_+)]|$ (for simplicity, let us denote these terms as (1) *performance* and (2) *bias* —or *fairness* under maximisation of the additive inverse— respectively; in turn, f must be found under a trade-off constraint between the two optimisation terms, given a performance-fairness trade-off coefficient $\Theta \in [0, 1]$ set a priori, such that $f^* = \text{argmax}_{f \in F}[\Theta \cdot performance + (1 - \Theta) \cdot fairness]$.

3 Related Work

Fairness in machine learning has garnered significant attention in recent years. This section covers measures of fairness in Section 3.1, and fair machine learning in Section 3.2.

3.1 Measures of Fairness

Fairness measures assess equality across sensitive attribute groups with respect to some computable statistic, and can be placed into two categories: (1) threshold-dependent fairness measures; and (2) threshold-independent fairness measures. We note that, regardless of threshold category, fairness measures have their values in the range $[0, 1]$, in which 0 translates to complete *fairness* while 1 indicates total *bias*.

Threshold-dependent fairness measures rely on class predictions, induced by a decision threshold t onto the model output score Z, rather than directly using Z. Prominently, we mention three such measures: (1) demographic parity; (2) equal opportunity; and (3) equalised odds. First, demographic parity requires the proportion of positively predicted instances to be equal across groups (Dwork et al., 2011); formally, $P(\hat{Y}{=}y_+|S{=}s_+) = P(\hat{Y}{=}y_+|S{=}s_-)$. As a measure of fairness, it is computed as the absolute difference between the two terms: $|P(\hat{Y}{=}y_+|S{=}s_+) - P(\hat{Y}{=}y_+|S{=}s_-)|$. Second, equal opportunity accounts for fairness with respect to model performance, by focusing on the true positive rate, rather than solely on the positive class predictions, ensuring equality of true positive rates between the sensitive attribute groups (Hardt et al., 2016); formally, $P(\hat{Y}{=}y_+|Y{=}y_+, S{=}s_+) = P(\hat{Y}{=}y_+|Y{=}y_+, S{=}s_-)$. To serve the purpose of fairness measure, it is computed as the absolute difference: $|P(\hat{Y}{=}y_+|Y{=}y_+, S{=}s_+) - P(\hat{Y}{=}y_+|Y{=}y_+, S{=}s_-)|$. Lastly, the measure of equalised odds extends the model performance considerations of equal opportunity by adding the two requirements of (1) equal false positive rates across the sensitive attribute groups, and (2) equal true positive rates and false positive rates (Hardt et al., 2016); formally, $P(\hat{Y}{=}y_+|Y{=}y_+, S{=}s_+) = P(\hat{Y}{=}y_+|Y{=}y_+, S{=}s_-) = P(\hat{Y}{=}y_+|Y{=}y_-, S{=}s_+) = P(\hat{Y}{=}y_+|Y{=}y_-, S{=}s_-)$. Computationally, it is given as: $|[|P(\hat{Y}{=}y_+|Y{=}y_+, S{=}s_+) - P(\hat{Y}{=}y_+|Y{=}y_+, S{=}s_-)|] - [|P(\hat{Y}{=}y_+|Y{=}y_-, S{=}s_+) - P(\hat{Y}{=}y_+|Y{=}y_-, S{=}s_-)|]|$. Ultimately, these threshold-dependent measures of fairness are ill-suited to address the current maritime use-case under our stipulated requirements (see Section 2), as the values of fairness may vary significantly depending on the selection of decision threshold.

Threshold-independent fairness measures do not rely on a decision threshold. Instead, they consider the model output score Z. The measure of strong demographic parity ensures that the output of a classification model is, on average, independent of sensitive attributes regardless of the decision threshold used (Jiang et al., 2019); formally, $P[(Z|S{=}s_+) > (Z|S{=}s_-)] = P[(Z|S{=}s_-) > (Z|S{=}s_+)]$. As a measure, strong demographic parity is calculated as the absolute difference $|P[(Z|S{=}s_+) > (Z|S{=}s_-)] - P[(Z|S{=}s_-) > (Z|S{=}s_+)]|$, simplifying to $|2 \cdot P[(Z|S{=}s_+) > (Z|S{=}s_-)] - 1|$ (Barata et al., 2024). The probability term is equal to the Receiver Operating Characteristic Area Under the Curve (ROC-AUC): $AUC(Z, S) = P[(Z|S{=}s_+) > (Z|S{=}s_-)]$ (Mason and Graham, 2002), and computed as the traditional classification performance $AUC(Z, Y) = P[(Z|Y{=}y_+) > (Z|Y{=}y_-)]$, by replacing Y with S. These mea-

sures align perfectly with our stipulated threshold-independence requirements.

3.2 Fair Machine Learning

The application of fairness in machine learning is the process by which inherent data biases are minimised within the final learned model. As described in Section 1, fairness processes may be categorised as either (1) *pre-* (2) *post-* or (3) *in*-processing, related to when in the learning pipeline (before, after, or during, respectively), fairness is enforced. The decision of which approach to select heavily depends on the level of algorithmic and data freedom of access and manipulation.

First, pre-processing involves the most freedom within the data themselves and minimises model bias by applying data transformations prior to model learning. These transformation learn a new representation of the data in which fairness is ensured while still preserving the fidelity of the task (Caton and Haas, 2024). Feldman et al. (2015) discuss an approach to remove information about sensitive attributes from a set of numeric covariates, by transforming the distribution of the variable towards the median while still retaining the rank order for instances. In practise, however, it has been shown that pre-processing methods generate classification models that still exhibit substantial bias (Agarwal et al., 2018).

Second, post-processing assumes that the degree of freedom is severely limited to solely the output of an already learned model. In other words, it takes into consideration that different sensitive attribute groups should have, for instance, unique decision thresholds to decide whether or not their predicted class is positive. This method is, however, not always appropriate since the usage of a sensitive attribute (either directly or indirectly (Petersen et al., 2021)) to decide upon an outcome may be, case-dependent, unethical in nature.

Lastly, in-processing assumes full control over the algorithmic development; i.e., the learning process via the arbitrary manufacturing of the learner. Different approaches exist to achieve this goal, of which we mention three of the most prominent in the literature: (1) constraint optimisation, (2) regularisation, and (3) adversarial learning. Constraint optimisation involves imposing additional (fairness) constraints to the objective function of the learner, in which the imposed constraints must be solvable under a convex optimisation problem definition; these constraints are usually formalised as either *weak* — such as the correlation between output and sensitive attribute (Kamishima et al., 2012) — or *strong*, as is the case with demographic parity and equalised odds (Agarwal et al., 2018). Regularisation approaches add penalty terms that include notions of fairness in the loss function, to reduce discrimination; this may be done, for example, in the form of a compound splitting criterion in a decision tree architecture (Barata et al., 2024), in which a split will be selected during learning based on a joint consideration of performance and fairness. Adversarial learning uses an adversary layer to deliver feedback on whether the training process is fair, and this feedback is used to improve the model; Zhang et al. (2018) present a framework where the adversary penalises the model if the sensitive attribute is predictable from the model output, encouraging the model to reduce bias and improve fairness. Since we have full control over the algorithmic decision-making in this work, and provided their prevalence and well-established performance in literature, our solution will be driven by in-processing methods.

4 Methods

This section covers the (fair) machine learning methods that we utilise in our experiments, with the purpose of finding the best setup for our current problem. In Section 4.1 we discuss the (fair) classification algorithms, and in Section 4.2 we discuss fairness through hyperparameter optimisation.

4.1 (Fair) Algorithms

In our methodology, we implemented both *traditional* and *fair* machine learning algorithms. We chose three learning algorithms known to perform well on classification tasks and their fair counterparts, following the aforementioned in-processing approaches: (1) Logistic Regression and Fair Logistic Regression (constraint optimisation, Section 4.1.1), (2) Random Forest and Fair Random Forest (regularisation, Section 4.1.2), and (3) Neural Network and Fair Adversarial Learning (adversarial learning, Section 4.1.3).

4.1.1 Logistic Regression & Fair Logistic Regression

The traditional Logistic Regression models the log-odds of a label as the linear combination of the available features (McCullagh and Nelder, 1989). Therein, the model extends to the fair case via a reductions approach to fair classification (Agarwal et al., 2018). It treats the underlying classifier (i.e. Logistic Regression) as a black box, in which the loss function is subject to fairness constraints which can be reduced to a sequence of cost-sensitive classification problems. The solutions to these cost-sensitive classification problems yield a classifier with the lowest error, subject to the desired fairness constraints.

The fairness constraints can be expressed as linear inequalities, using the definition of fairness most suitable to the use-case problem, so long as it falls under a solvable convex optimisation problem (e.g., demographic parity). To this end, Lagrange multipliers are introduced to handle these constraints. Lagrange multipliers are used as a mathematical technique to find extreme values of a function under constraints. The idea is to convert the function under constraints to a form such that the derivative test of a function that is not under constraints can still be applied. The Lagrange multiplier scales the gradient of the constraint so that the gradient of the objective function and the gradient of the constraint function are of the same magnitude. Therein, a new function (*Lagrange function*) is generated which, by setting all partial derivatives to zero, ensures that an extreme point of the objective function is found under the stipulated constraints.

The Fair Logistic Regression tries to minimise the classification error using a cost-sensitive classifier, while adjusting the Lagrange multipliers (which can be seen as the penalties for unfairness) to ensure the fairness constraints are respected, until an equilibrium is reached. This equilibrium corresponds to the balance between performance and fairness, determined by a *constraint weight* parameter set a priori, which stipulates the relative importance put on the constraint violation when selecting the best model, with *constraint weight* $\in [0, 1]$ where *constraint weight* $= 0$ promotes classification performance and *constraint weight* $= 1$ promotes fairness.

4.1.2 Random Forest & Fair Random Forest

A traditional Random Forest algorithm fits multiple decision tree classifiers with a specific splitting criterion (traditionally *Gini* or *Entropy*) onto subsets of the original dataset, averaging the output across all decision tree outputs to generate the output of the Random Forest model (Breiman, 2001). Each tree will split its assigned sub-dataset into smaller partitions recursively, so long as the new data split increases the splitting criterion score comparatively to the score of the node from which the split originated.

The Fair Random Forest (Barata et al., 2024) works similarly to its traditional counterpart, but instead uses *fair* tree classifiers, which take the Gini index splitting criterion and add to it the fairness component corresponding to the strong demographic parity, termed Splitting Criterion AUC For Fairness (SCAFF), given in the following Eq. 1:

$$SCAFF(Z, Y, S, \Theta) = (1 - \Theta) \cdot |2 \cdot AUC(Z, Y) - 1| - \Theta \cdot |2 \cdot AUC(Z, S) - 1| \quad (1)$$

Here, $AUC(Z, Y)$ —the classification performance term— and $AUC(Z, S)$ —the fairness term— are the ROC-AUC computed as per Section 3.1. The terms are then scaled to account for the symmetry of the binary AUC case; i.e., a split of $AUC = 0.1$ is as optimal as a split of $AUC = 0.9$. Finally, SCAFF incorporates an elastic-net-like linear combination of performance and fairness terms via the parameter $\Theta \in [0, 1]$, controlling for an allowed performance-fairness trade-off, in which $\Theta = 0$ promotes classification performance and $\Theta = 1$ promotes fairness.

4.1.3 Neural Network & Fair Adversarial Learning

A traditional Neural Network classification model consists of layers of neurons that apply non-linear transformations to weighted sums of its inputs (Hinton, 1989). The model is learned via backpropagation towards towards the minimisation of a loss function, traditionally cross-entropy, under which optimisation algorithms (e.g., Stochastic Gradient Descent) adjust the weights of the network, until the loss converges.

Fair Adversarial Learning modifies this process by introducing an adversarial network on top of the final output layer (Zhang et al., 2018). In this approach, two networks are trained simultaneously: the primary neural network, which focuses on classification accuracy, and an adversarial network that tries to predict the sensitive attributes from the classifier outputs. The classifier learns to minimise both its classification error and a fairness penalty that reflects how well the adversary can predict the sensitive attributes.

Succinctly, the predictor network has loss term L_P (the error towards Y) and network weights W, while the adversary network has loss term L_A (the error towards S), and network weights U. While weights U are updated to minimise L_A according to the gradient $\nabla_U L_A$ weights W are updated to minimise L_P, and are updated via $\nabla_W L_P - \text{proj}_{\nabla_W L_A} \nabla_W L_P - \alpha \nabla_W L_A$. Accordingly, the term $\text{proj}_{\nabla_W L_A} \nabla_W L_P$ prevents the predictor from moving in a direction that helps the adversary decrease its loss, while the term $\alpha \nabla_W L_A$ attempts to increase the loss of the adversary. Finally, $\alpha \in (0, 1]$ is a hyperparameter which controls the balance between accuracy and fairness; a higher α penalises unfairness more (promoting fairness), while a lower value places more emphasis on classification performance.

4.2 Fairness Through Hyperparameter Optimisation

Unfairness emerges when optimising classification models solely for predictive performance (Cruz et al., 2021). From the literature, fairness-aware hyperparameter optimisation attempts to mitigate this unfairness via a general model-agnostic fairness-aware objective function given as the following Eq. 2:

$$g(\lambda) = \Theta \cdot \rho(\lambda) + (1 - \Theta) \cdot \phi(\lambda) \tag{2}$$

Here, $\lambda \in \Lambda$ is a hyperparameter configuration, g is the objective function we wish to maximise, ρ is a predictive performance function, ϕ is a fairness function, and Θ is a weighting parameter which determines the relative importance of predictive performance and fairness. In this general function, ρ and ϕ can be replaced with custom performance and fairness functions. We note that g is functionally identical to the optimisation term defined in our requirements used to find the optimal classification model f^* (see Section 2).

Explicitely, given a learning algorithm $h \in H$ with hyperparameters $\lambda \in \Lambda$ which generates a classification model f_{h_λ} under data input X, Y, and S, with output Z, a performance term $AUC(Z, Y)$, a fairness term $AUC(Z, S)$, and a performance-fairness trade-off parameter value $\Theta \in [0, 1]$ set a priori, we find the optimal classification model f^* via:

$$f^* = \underset{f_{h_\lambda} \in F}{\operatorname{argmax}}[(1 - \Theta) \cdot AUC(Z, Y) - \Theta \cdot AUC(Z, S)^*], \tag{3}$$

$$AUC(Z, Y) = P[(Z|Y_{=}y_+) > (Z|Y_{=}y_-)], \tag{4}$$

$$AUC(Z, S)^* = 0.5 + |AUC(Z, S) - 0.5|, \tag{5}$$

$$AUC(Z, S) = P[(Z|S_{=}s_+) > (Z|S_{=}s_-)], \tag{6}$$

$$f_{h_\lambda} = h_\lambda(X, Y, S),\ h \in H,\ \lambda \in \Lambda, \tag{7}$$

$$Z = f_{h_\lambda}(X) \tag{8}$$

We use a compound hyperparameter optimisation criterion (i.e., objective function) which is defined as a linear combination between fairness and classification performance measures as per the splitting criterion in Eq. 1 (Section 4.1). Note that $AUC(Z, S)^*$ represents the fairness measure strong demographic parity, adjusted to range $[0.5, 1]$ where any original value below 0.5 gets "mirrored" at 0.5; e.g., 0.4 becomes 0.6. In doing so, we ensure that both performance and fairness terms are optimized within the same target range, ensuring that the impact of increasing/decreasing values of Θ is linear and thus more intuitive for the end user.

5 Data

This section addresses the data used in our experiments. In addition to the data provided by the ILT, a benchmark fairness dataset was used. Section 5.1 focuses on the characteristics of the ILT dataset, while Section 5.2 describes the benchmark in question.

5.1 ILT

The ILT dataset consists of 423 instances (NGO Shipbreaking Platform, 2024; Lloyd's List intelligence, 2024). Each instance represents a ship that has either been appropriately dismantled according to legislation or illegally beached, which is the target variable of interest. Notably, no ships currently sailing are included in this dataset; it only contains ships that have definitively reached their end-of-life. The dataset includes 11 variables, comprising both the target variable and the sensitive attribute. They constitute a mixture of numerical and categorical variables. For security reasons, we are not disclosing the variables used specifically.

The relationship between target and sensitive attributes is displayed in Table 1. It can be observed that of all the ships in the dataset sailing under a flag of convenience, 83.17% were beached. Of all the ships not sailing under a flag of convenience, 59.94% were beached. Additionally, 30.32% of all the beached ships sailed under a flag of convenience, while 11.64% of the non-beached ships sailed under a flag of convenience.

Table 1: The numbers of ships in the dataset that were (not) beached and (not) a flag of convenience.

	Flag of convenience	*No flag of convenience*	Total
Beached	84	193	277
Not beached	17	129	146
Total	101	322	423

While the dataset provides valuable insights, its relatively small size of 423 instances poses certain limitations, especially for the current supervised fair machine learning task. It is generally accepted that small sample sizes may lead to overfitting, where the model performs well on the training data but poorly on unseen data. This issue is particularly pronounced in fair classifier learning, where the goal is to ensure that the model's decisions are unbiased across different groups. A specific challenge arises from the very low number of instances (see Table 1) where the target variable is negative (not beached) and the sensitive attribute is positive (flag of convenience). In this dataset, there are only 17 such instances. This scarcity significantly undermines the model's learning potential. With such few examples, the model may struggle to accurately learn the relationship between these variables, leading to poor performance-fairness trade-offs.

Towards methodological assurance under this caveat, an additional fairness benchmark dataset is used to validate our design choices.

5.2 Benchmark

In addition to the primary dataset of ships, the Adult dataset from the UCI Machine Learning Repository (Becker and Kohavi, 1996) is used as a benchmark in the experimental setup. This dataset is widely recognised in the field of machine learning and is frequently utilised for studying fairness.

The benchmark dataset consists of $30,913$ instances, with 12 attributes, and is used to predict whether an individual's annual income exceeds $\$50,000$ based on census data. The target variable is binary, indicating whether income is greater than \$50K (positive) or not (negative). The dataset includes a mix of continuous and categorical variables, such as age, education level, occupation, and gender, which are essential for various machine learning and fairness studies. In our case, gender was selected as the sensitive attribute as it is encoded binarily, just as the sensitive attribute for our use-case dataset.

The larger sample size of the Adult dataset ($30,913$ instances) significantly alleviates the caveats associated with the relatively small ships dataset (423 instances), helping to assure methodological quality, and providing a robust foundation for our experimental design. Moreover, the sensitive attribute and class relationship imbalances are also mitigated.

The relationship between target and sensitive attributes is displayed in Table 2. It can be observed that of all the people in the dataset that were female, 10.96% had an annual income greater than \$50K. Of all the people that were male, 30.79% had an annual income greater than \$50K. Additionally, 14.90% of all the people that had an annual income greater than \$50K were female, while 38.74% of the people that did not have an annual income greater than \$50K were female.

Table 2: The numbers of people in the Adult dataset whose annual income was (not) greater than \$50K and who were (fe)male.

	Female	*Male*	Total
Greater than $\$50K$	$1,117$	$6,381$	$7,498$
Not greater than $\$50K$	$9,070$	$14,345$	$23,415$
Total	$10,187$	$20,726$	$30,913$

6 Experiments

This section will cover the preprocessing steps in Section 6.1, the algorithms in Section 6.2, the measures in Section 6.3, and detail the experimental setup in Section 6.4.

6.1 Preprocessing

The dataset preprocessing involves four steps to prepare the data such that it may be used in our experiments: (1) feature scaling to standardise features, (2) handling of missingness, (3) categorical aggregation, and (4) one-hot encoding. First, all numerical columns were scaled using the median and the quantile range, as to ensure more robust scaling than when using the mean and standard deviation. The median value is removed from the observations and the result is divided by the inter-quantile range. The scaling did not consider missing values when computing the statistics with which scaling was performed. Second, missing values of the numerical columns were systematically managed by using binary indicators for their identification and mean imputation for their handling, providing a balanced approach to preserving data integrity. For categorical columns, missing values were indicated using the category "missing". Third, all categorical columns were clamped such that rare categories were aggregated into one, so as to avoid a lot of low-frequency categories. This was done by firstly computing the frequency of each unique category. Based on these frequencies, the relative frequency of each category was calculated. A threshold is set as the reciprocal of the number of unique categories. This threshold represents an idealised relative frequency that would result if all categories were perfectly evenly distributed. Next, the index was identified of the category in the list of categories (sorted in descending order based on frequency) where the absolute difference between the relative frequency and the threshold was minimised, implying the most evenly distributed frequency up to that index. A visualisation can be found in Appendix A. Lastly, the categories were one-hot encoded. This is a technique that is used to represent categorical variables as binary values in a machine learning model. The categorical variables will prepare separate columns for all unique values that occur, denoting each instance as a "0" or a "1" for these columns.

6.2 Algorithms

We implemented both *traditional* and fair machine learning algorithms, such that for each traditional algorithm, a fair algorithm counterpart is applied.

On the one hand, three traditional approaches were selected: (1) Random Forest (Breiman, 2001), (2) Logistic Regression (McCullagh and Nelder, 1989), and (3) Neural Network (Hinton, 1989). The Random Forest classifier and the Logistic Regression classifier were implemented using the Scikit-learn Python package (Pedregosa et al., 2011), while the Neural Network classifier was implemented using the AI Fairness 360 Python package (Bellamy et al., 2018). These approaches serve as baselines, providing a reference point for evaluating classification performance.

On the other hand, the three fair counterparts considered were, logically: (1) Fair Random Forest (Barata et al., 2024), (2) Fair Logistic Regression (Agarwal

et al., 2018), and (3) Fair Adversarial Learning (Zhang et al., 2018). The Fair Random Forest was implemented using the fair trees Python package (Barata et al., 2024), while the Fair Logistic Regression and Fair Adversarial Learning were implemented using the AI Fairness 360 Python package (Bellamy et al., 2018). These fair machine learning algorithms were selected for their ability to incorporate fairness constraints while maintaining adequate performance in classification tasks.

By implementing both traditional and fair algorithms, this approach allows for a thorough comparative analysis, offering insights into the trade-offs involved in applying fairness constraints.

6.3 Measures of Performance and Fairness

Throughout our entire experimental design, (1) the measure of performance ROC-AUC, and (2) the strong demographic parity measure of fairness were used. Their selection was based on our requirements, as stipulated in Section 2. First, the ROC-AUC can be interpreted as the probability that a model ranks a randomly-selected positive instance more highly than a randomly-selected negative instance (see Section 3.1); i.e., it is calculated using the model's outcome scores, thereby circumventing the need for a choice of label-inducing decision threshold, as per our requirements. Under expected conditions, ROC-AUC $\in [0.5, 1]$, in which the greater the ROC-AUC value, the greater the classification performance of a given model.

Second, the strong demographic parity can be interpreted as the ROC-AUC, but with respect to the sensitive attribute, rather than the target variable. As such, it too is a threshold-independent measure. The strong demographic parity $\in [0, 1]$, in which the smaller the value, the lower the bias and, conversely, the greater the fairness of the model in question; i.e., it is technically a measure of bias.

To ensure robust and generalisable expected measures of performance and fairness, including hyperparameter optimisation, 10-fold nested cross validation (CV) was used in all our experiments. Nested CV involves two layers: (1) an outer loop that computes the expected performance and fairness measures, and (2) an inner loop that identifies the best set of hyperparameters for training a model and evaluating the performance on the test set of the corresponding outer fold. This structure ensures no leakage or overfitting, as no training and test sets overlap. Hyperparameter optimisation was conducted using the Hyperopt Python package, which uses the Tree-structured Parzen Estimator algorithm to find an optimal set of hyperparameters (Bergstra et al., 2013). This is a sequential model-based optimisation approach. The maximum number of evaluations for Hyperopt was set to 100 as it should provide adequate optimisation within a reasonable computation budget. An exhaustive list of all the hyperparameters that were optimised for each algorithms and their ranges can be found in Appendix B.

When testing results for significance, p-values are obtained from two-tailed independent t-tests.

6.4 Experimental Setup

Four distinct experimental setups were considered, each focusing on different combinations of traditional and fair algorithms with the specific criteria used for hyperparameter optimisation: (1) traditional machine learning algorithms optimised for classification performance, (2) fair machine learning algorithms optimised for classification performance, (3) traditional machine learning algorithms optimised for both classification performance and fairness simultaneously and (4) fair machine learning algorithms optimised for both classification performance and fairness simultaneously.

The first setup corresponds to the traditional approach with which to compute expected performance under a standard classification problem. Its purpose is to generate benchmark performance and fairness measures with which we can compare the following fair machine learning experimental setups.

For the remaining setups, a range of fairness parameter values are inspected to assess the resulting performance-fairness trade-off: $\Theta \in [0, 1]$ in Fair Random Forest, *constraint weight* $\in [0, 1]$ in Fair Logistic Regression, and $\alpha \in [0, 0.1]$ in Fair Adversarial Learning. For Θ and *constraint weight*, values range from 0 up to and including 1 in increments of 0.1, while for α, values range from 0 up to and including 0.1 in increments of 0.01. The second experimental setup sets each fairness parameter a priori. Subsequently, model learning ensues whilst optimising for classification performance exclusively.

Finally, the remaining experimental setups do not set the fairness parameters a priori, but rather allow them to be optimised just as if they were any other hyperparameter. The hyperparameters are then optimised using fairness-aware hyperparameter optimisation. Performance-fairness trade-off parameter Θ in our compound criterion is evaluated in our experiments in increments of 0.1, from 0 and up to and including 1. The third experimental setup combines traditional machine learning with fairness-aware hyperparameter optimisation and the fourth experimental setup combines fair machine learning with fairness-aware hyperparameter optimisation.

Every setup described was applied to two datasets in question: (1) ILT, and (2) Adult. The sensitive attribute is not used as a predictor in any of the experiments. For reproducibility, our experiments can be found in `https://github.com/LWijne/Predicting-illegal-ship-breaking-via-fairness-aware-classifier-optimisation.git`.

7 Results

In this section, we will outline the results of the experiments that were conducted to answer the problem statement. The results are organised according to the research questions outlined in Section 1. Section 7.1 covers the predictive performance and fairness of the traditional machine learning algorithms in relation to the ILT and the Adult datasets. In Section 7.2 we will discuss the results of setting the fairness parameter of the fair machine learning algorithms a priori. In Section 7.3 we will discuss the results of using fairness-aware hyperparameter optimisation. In Section 7.4 we will compare all results.

7.1 Traditional Machine Learning Algorithms

The results of our experiments with respect to the traditional classification algorithms can be seen in Figure 1. It depicts the ROC-AUC (vertical axis) and strong demographic parity (horizontal axis) values for each of the three algorithms (green, red, and blue) applied to the ILT (left) and Adult (right) datasets.

For both datasets, it can be observed that Random Forest achieves the highest average ROC-AUC while also exhibiting the highest strong demographic parity scores, indicating higher classification performance than the other two algorithms but also lower fairness. Logistic Regression performs the worst across the three algorithms on both datasets, with an ROC-AUC score significantly lower than Random Forest on the ILT dataset ($p = 0.0003$), and significantly lower than Random Forest ($p < 0.0001$) and Neural Network ($p < 0.0001$) on the Adult dataset. Logistic Regression does perform better than the other two in terms of fairness, with a strong demographic parity score significantly lower than Random Forest on the ILT dataset ($p = 0.0093$), and significantly lower than Random Forest ($p < 0.0001$) and Neural Network ($p < 0.0001$) on the Adult dataset. Neural Network is in between Random Forest and Logistic Regression in terms of both predictive performance and fairness.

With respect to the ILT dataset, both measures of classification performance and fairness show more variation for Neural Network and Logistic Regression than for Random Forest, suggesting that Random Forest might be more stable on both accounts. The results on the ILT dataset show more variation than the results on the Adult dataset. As discussed in Section 5, the ILT dataset is relatively small and on top of that has imbalanced classes, contributing to greater variation in the results as the training data for each fold in the CV might not be sufficiently representative of the overall dataset.

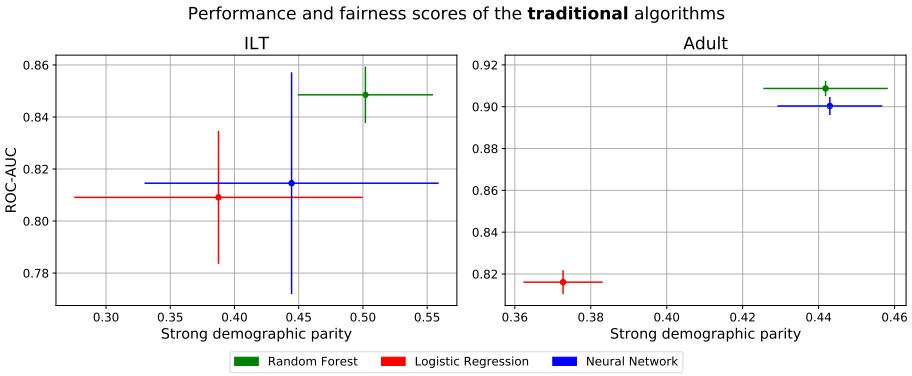


Figure 1: ROC-AUC and strong demographic parity scores across 10-fold nested CV of traditional machine learning algorithms on the ILT (left) and Adult (right) datasets.

7.2 Fair Machine Learning Algorithms

Here, we investigate the impact of fair machine learning algorithms and how varying their fairness parameters influences their performance and fairness. Figure 2 shows this influence across different evaluations of their respective fairness parameters. It depicts the ROC-AUC and strong demographic parity (both on the vertical axis) for different fairness parameter values (horizontal axis) values for each of the three algorithms (green, red, and blue) applied to the ILT (left) and Adult (right) datasets.

For Fair Random Forest, it is clear that adjusting the fairness parameter Θ influences both performance and fairness for both datasets, with the strong demographic parity reaching 0.0 for $\Theta = 1$. For Fair Logistic Regression, the impact that shifting the fairness parameter has on classification performance and fairness is far lesser than for Fair Random Forest for these two datasets, although still visible. For Fair Adversarial Learning, adjusting the fairness parameter α influences fairness more than it influences performance on both datasets.

The results on the ILT dataset show more variation for all of the fair algorithms, due to the relatively small size of the dataset and the imbalanced classes, as mentioned before.

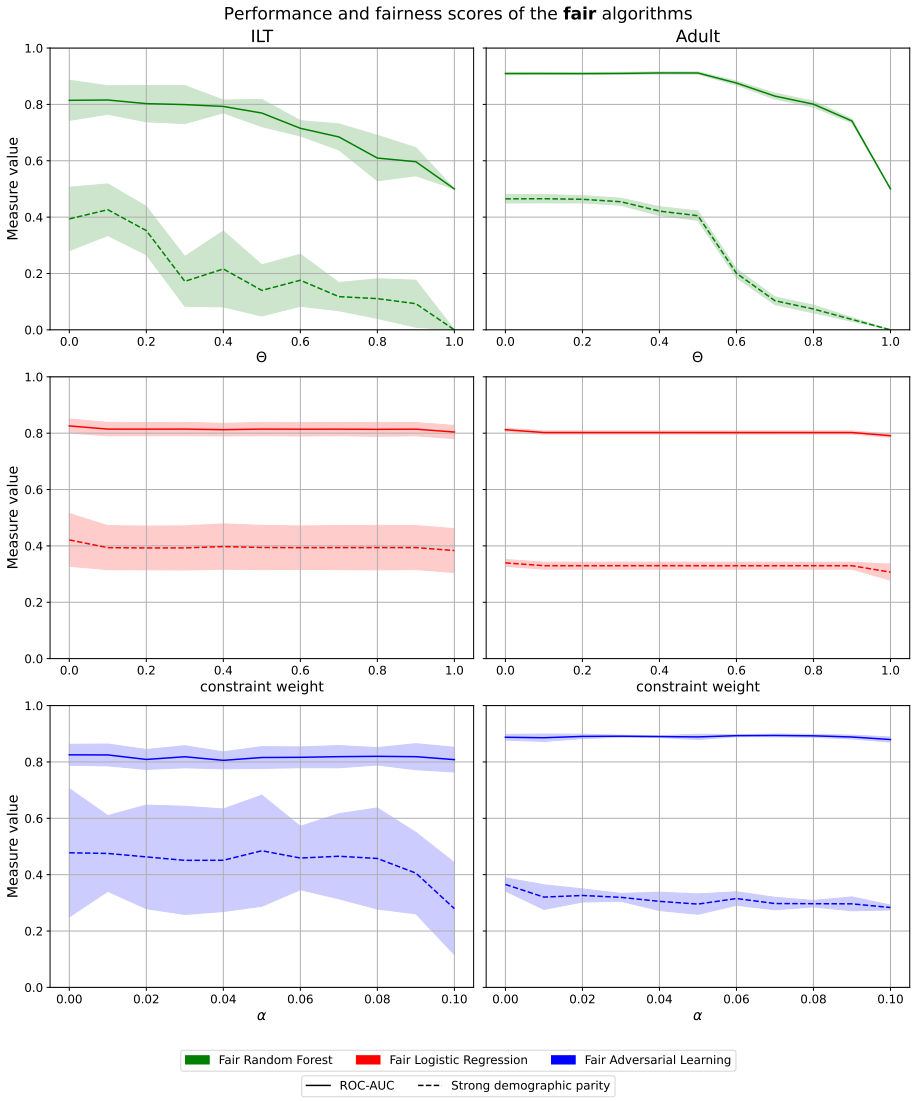


Figure 2: ROC-AUC and strong demographic parity scores of distinct fairness parameters across 10-fold nested CV of fair machine learning algorithms on the ILT and Adult datasets.

To gain further insight into the trade-off between performance and fairness for all of these algorithms, Figure 3 shows the pair-wise ROC-AUC and strong demographic parity values at each sequential fairness parameter evaluation. It depicts the ROC-AUC (vertical axis) and strong demographic parity (horizontal axis) values for each of the three algorithms (green, red, and blue) applied to the ILT (left) and Adult (right) datasets.

It is clearly visible that Fair Random Forest has the largest performance-fairness trade-off range. For both datasets, Fair Random Forest achieves much lower strong demographic parity scores than the other two fair machine learning

algorithms. In the case of the Adult dataset, it also dominates over the other two algorithms at any point. This is not the case for the ILT dataset, although the differences are small.

It is also clear that the algorithms produce more smooth, concave curves on the Adult dataset than on the ILT dataset. Again, this is most likely due to the relatively small size and imbalanced classes of the ILT dataset.

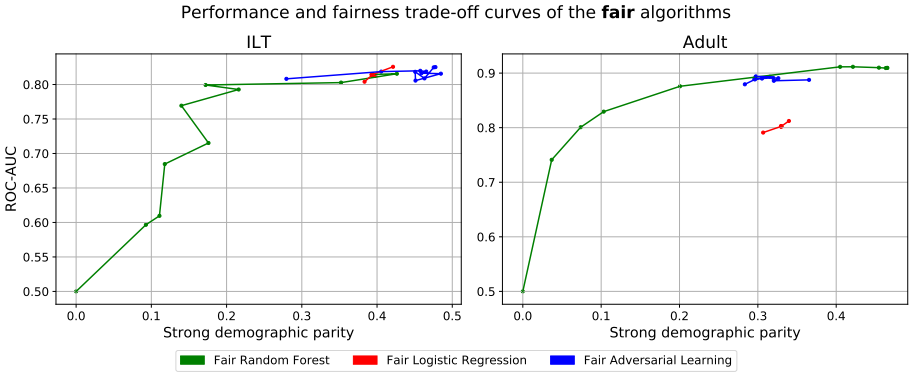


Figure 3: ROC-AUC scores plotted against strong demographic parity scores of distinct values of the fairness parameters across 10-fold nested CV of fair machine learning algorithms on the ILT (left) and Adult (right) datasets.

7.3 Fairness-aware Hyperparameter Optimisation

We now optimise the parameters found in Table 3 in Appendix B, along with the fairness parameters Θ, *constraint weight*, and α of the fair machine learning algorithms, for both classification performance and fairness simultaneously as set out in Section 6.4. Section 7.3.1 covers the results on the traditional algorithms, and Section 7.3.2 covers the results on the fair algorithms.

7.3.1 Traditional Machine Learning Algorithms

Figure 4 shows the influence of adjusting Θ using fairness-aware hyperparameter optimisation on the traditional algorithms. It depicts the ROC-AUC and strong demographic parity (both on the vertical axis) for different Θ values (horizontal axis) values for each of the three algorithms (green, red, and blue) applied to the ILT (left) and Adult (right) datasets.

Pairing all three algorithms with fairness-aware hyperparameter optimisation, Random Forest achieves the lowest unfairness scores on the Adult dataset, in line with the results from the original study on the Adult dataset, which also experimented with these three algorithms (Cruz et al., 2021). We see that Θ has the most impact on Random Forest, the least impact on Logistic Regression, and Neural Network is in between these two.

On the ILT dataset, higher values of Θ produce more unstable results in terms of both ROC-AUC and strong demographic parity.

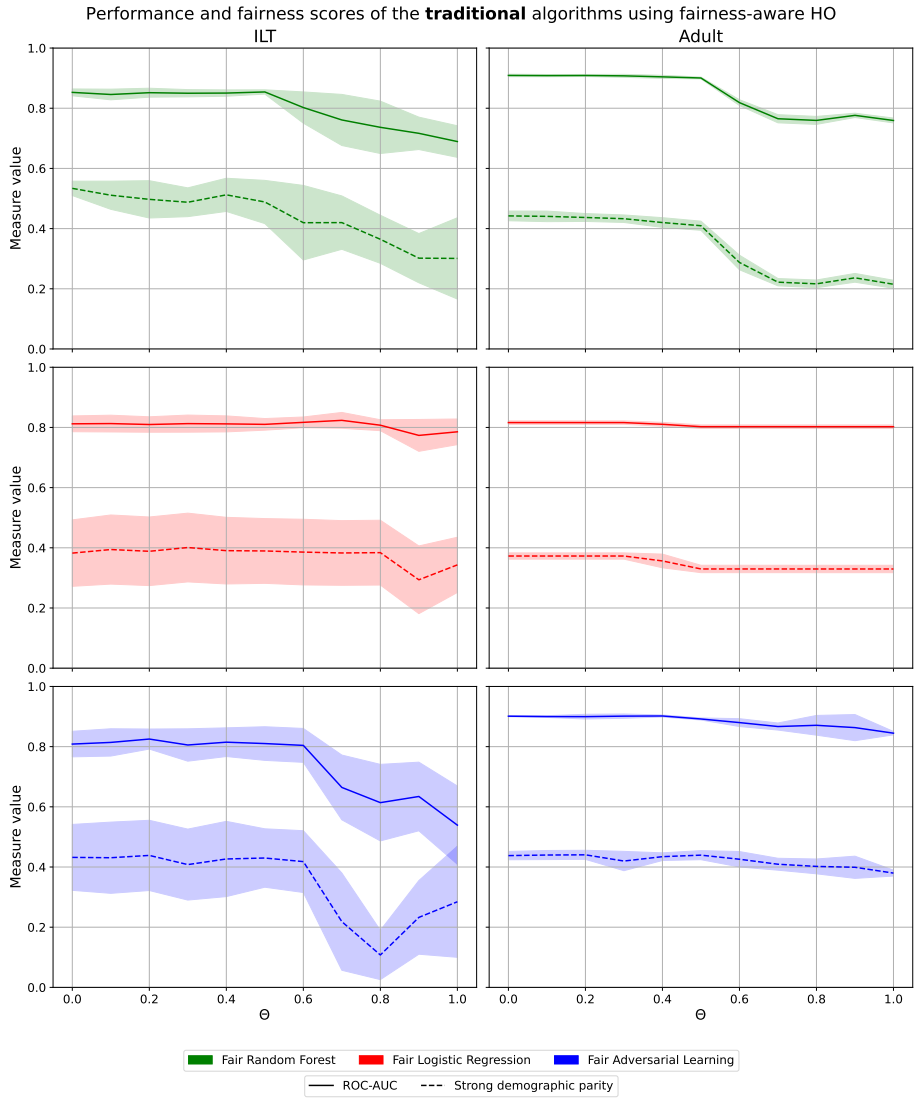


Figure 4: ROC-AUC and strong demographic parity scores of distinct values of Θ across 10-fold nested CV of traditional machine learning algorithms on the ILT and Adult datasets.

To gain further insight into the trade-off between performance and fairness for all of these algorithms, Figure 5 shows the pair-wise ROC-AUC and strong demographic parity values at each sequential Θ evaluation. It depicts the ROC-AUC (vertical axis) and strong demographic parity (horizontal axis) values for each of the three algorithms (green, red, and blue) applied to the ILT (left) and Adult (right) datasets.

Paired with fairness-aware hyperparameter optimisation, Logistic Regression shows the smallest performance-fairness trade-off range of these three algorithms. On the ILT dataset, Logistic Regression achieves higher ROC-AUC

scores than the other two algorithms at any point where those achieve the same fairness score. The lowest unfairness score is obtained by Neural Network. On the Adult dataset, Random Forest completely covers the Pareto-efficient frontier.

Again, the curves are a lot smoother on the Adult dataset compared to the ILT dataset.

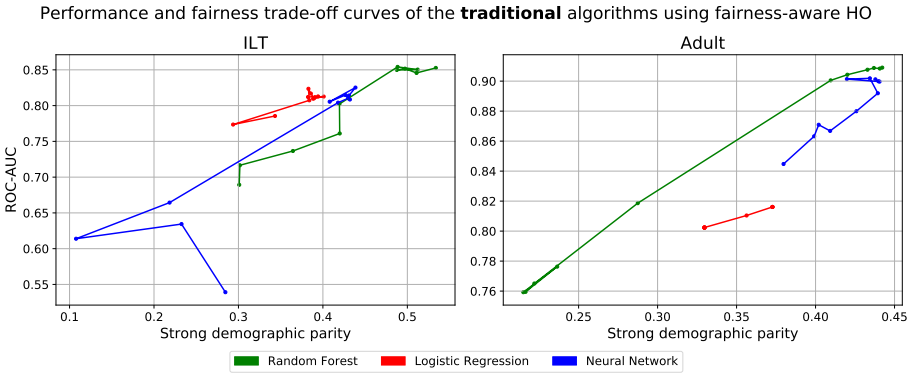


Figure 5: ROC-AUC scores plotted against strong demographic parity scores of distinct values of Θ across 10-fold nested CV of traditional machine learning algorithms on the ILT (left) and Adult (right) datasets.

7.3.2 Fair Machine Learning Algorithms

Figure 6 shows the influence of adjusting Θ using fairness-aware hyperparameter optimisation on the fair algorithms. It depicts the ROC-AUC and strong demographic parity (both on the vertical axis) for different Θ values (horizontal axis) values for each of the three algorithms (green, red, and blue) applied to the ILT (left) and Adult (right) datasets.

We notice that for Fair Random Forest, the plot overlaps with the one in Figure 2. This is due to the fact that the objective function is actually the same as SCAFF in Fair Random Forest. For Fair Logistic Regression, significantly lower unfairness scores are obtained in this manner, compared to purely adjusting the fairness parameters, for both the ILT dataset ($p < 0.0001$) and the Adult dataset ($p < 0.0001$). The same holds for Fair Adversarial Learning for the Adult dataset ($p = 0.0002$). As opposed to all other experiments, in this case all three algorithms reach strong demographic parity scores below 0.2 on both datasets.

The results on the ILT dataset show a lot more variation than the results on the much larger Adult dataset, especially in the case of Fair Adversarial Learning.

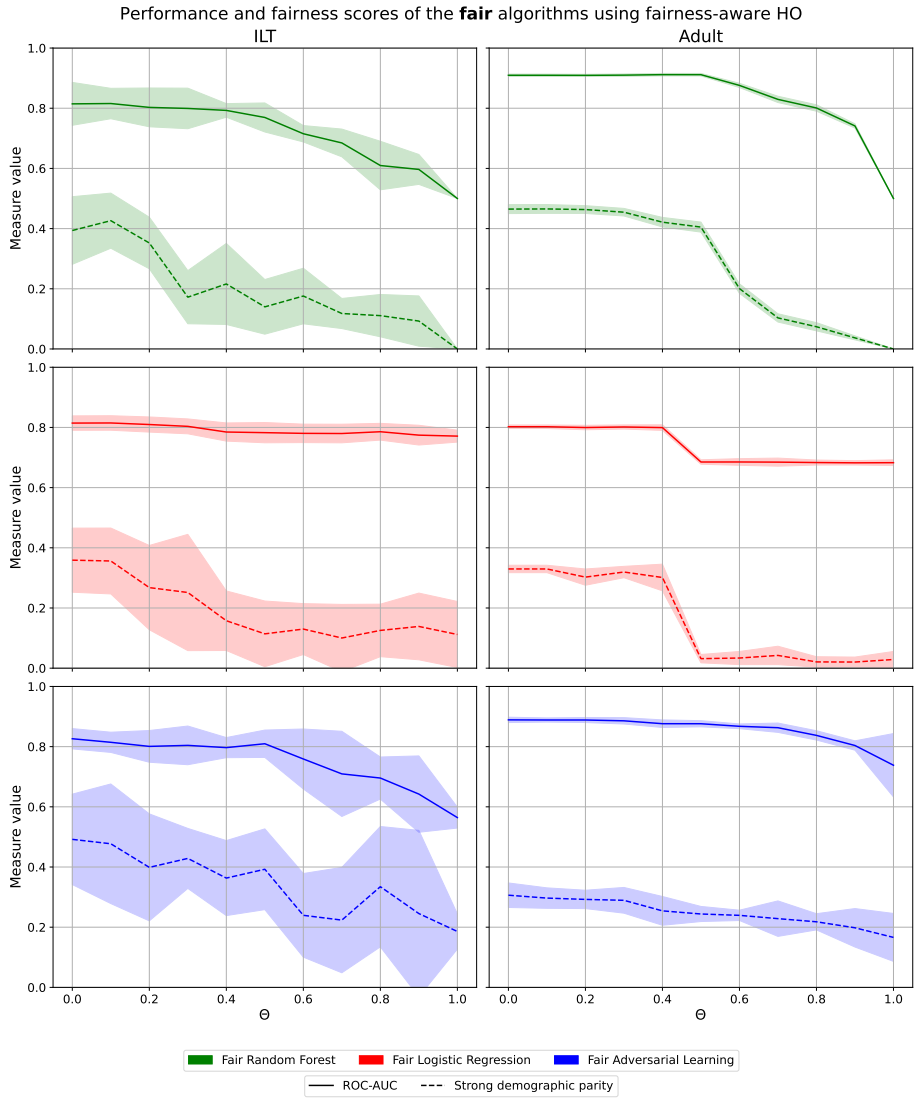


Figure 6: ROC-AUC and strong demographic parity scores of distinct values of Θ across 10-fold nested CV of fair machine learning algorithms on the ILT and Adult datasets.

To gain further insight into the trade-off between performance and fairness for all of these algorithms, Figure 7 shows the pair-wise ROC-AUC and strong demographic parity values at each sequential Θ evaluation. It depicts the ROC-AUC (vertical axis) and strong demographic parity (horizontal axis) values for each of the three algorithms (green, red, and blue) applied to the ILT (left) and Adult (right) datasets.

Fair Random Forest still has the biggest performance-fairness trade-off range, although the other two algorithms have greatly improved their ranges using fairness-aware hyperparameter optimisation. On the ILT dataset, Fair Adver-

sarial Learning achieves the highest performance score, Fair Random Forest achieves the lowest unfairness score, and the Pareto-efficient frontier is made up of points from all three algorithms. This means that based on the desired performance and fairness, all three algorithms could be most suitable in this case. On the Adult dataset, Fair Random Forest shows a really nice performance-fairness trade-off curve that dominates over the Fair Adversarial Learning curve. Fair Logistic Regression shows strong demographic parity scores of ≤ 0.1 while maintaining relatively high performance scores, on both datasets.

Again, the much larger Adult dataset has more smooth, concave curves.

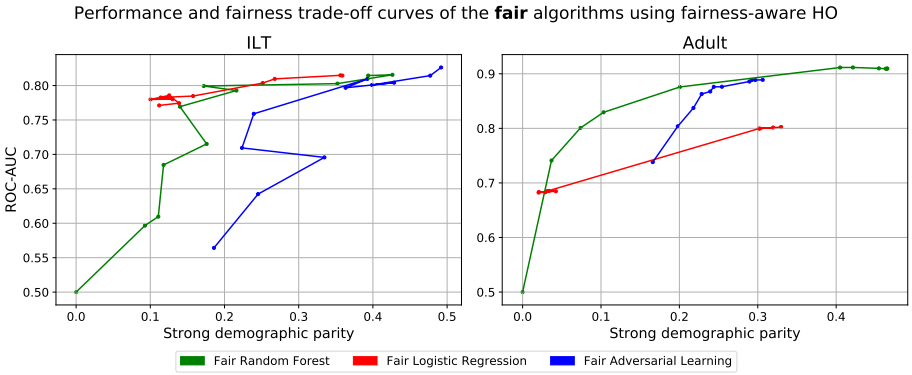


Figure 7: ROC-AUC scores plotted against strong demographic parity scores of distinct values of Θ across 10-fold nested CV of fair machine learning algorithms on the ILT (left) and Adult (right) datasets.

7.4 Comparing All Methods

Figure 8 depicts the ROC-AUC (vertical axis) and strong demographic parity (horizontal axis) values for each employed method (shades of green, red, and blue), with(out) fairness-aware hyperparameter optimisation (solid and dotted lines) applied to the ILT (left) and Adult (right) datasets.

The darker shades of green, red, and blue, respectively, represent the traditional methods. The lighter shades represent the fair methods. The solid lines represent algorithms that were optimised for performance. The dotted lines represent algorithms that were optimised for both classification performance and fairness simultaneously.

For (Fair) Random Forest, the two lightest shades overlap completely, since our objective function is identical to SCAFF in Fair Random Forest. For the ILT dataset, the traditional Random Forest achieves the highest performance, while Fair Random Forest achieves the lowest unfairness and a larger performance-fairness trade-off range. For the Adult dataset, Fair Random Forest completely covers the Pareto-efficient frontier.

For (Fair) Logistic Regression, remarkably Fair Logistic Regression paired with fairness-aware hyperparameter optimisation achieves much lower unfairness scores than "plain" Fair Logistic Regression, while still achieving decent classification performance scores.

For Neural Network and Fair Adversarial Learning, the traditional Neural Network achieves the lowest unfairness on the ILT dataset, although the results are unstable. On the Adult dataset, Fair Adversarial Learning achieves the lowest strong demographic parity scores while still maintaining relatively high ROC-AUC scores.

While all results on the ILT dataset show instability, the results on the Adult dataset consistently show that fair machine learning algorithms paired with fairness-aware hyperparameter optimisation achieve the lowest unfairness scores.

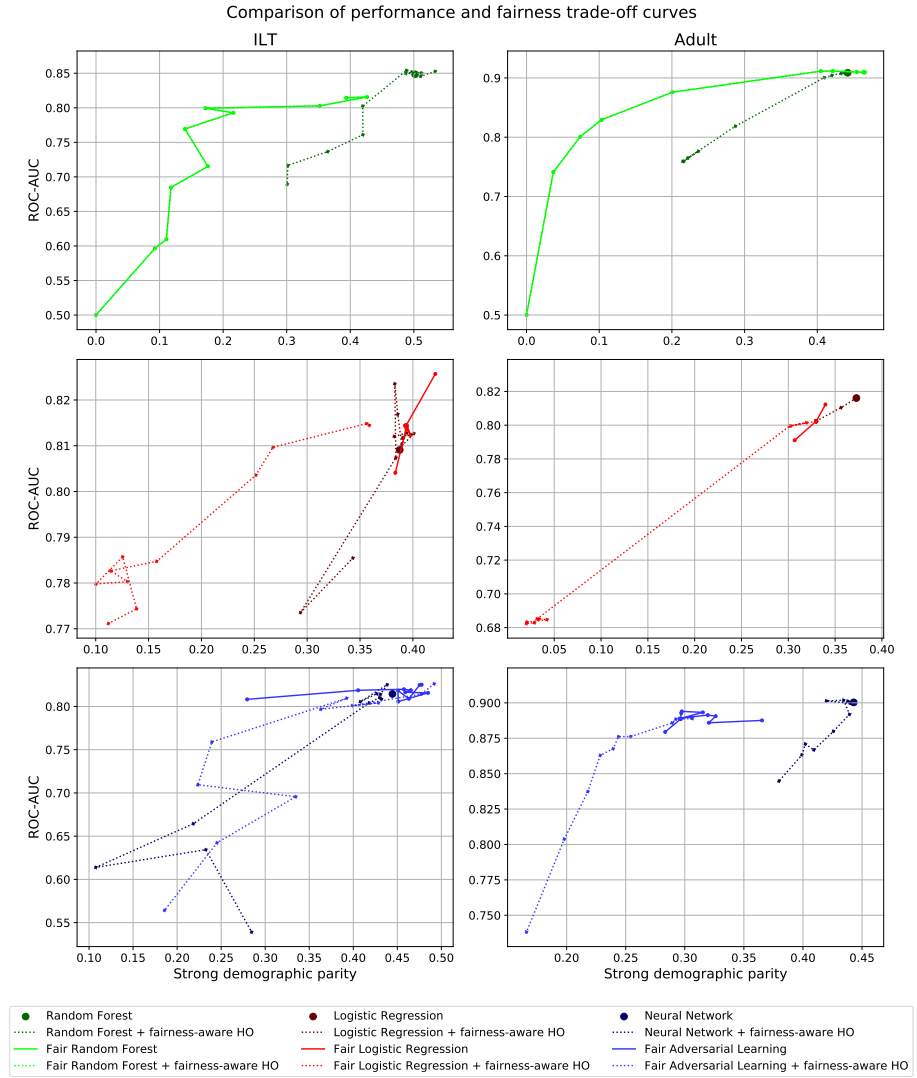


Figure 8: ROC-AUC scores plotted against strong demographic parity, showing performance-fairness curves for all methods using 10-fold nested CV on the ILT (left) and Adult (right) datasets.

8 Discussion

This section reflects on the findings presented in Section 7, discussing the implications, limitations, and potential directions for future work.

The results presented in this work demonstrate the complex interplay between fairness and predictive performance in (fair) classification models, and the difficulty that is added when dealing with a small dataset.

Random Forest was the highest performing in terms of classification performance, but showed the highest unfairness. The inherent low complexity of Logistic Regression might limit its performance on complex tasks (reflected in lower ROC-AUC scores) but simultaneously refrain it from exploiting biases that lead to unfairness (reflected in lower strong demographic parity scores).

We saw that pairing traditional machine learning with fairness-aware hyperparameter optimisation achieves much lower unfairness scores than with regular hyperparameter optimisation, and gives more control over the performance-fairness trade-off.

Experimenting with fair machine learning algorithms, Fair Random Forest exhibited a significant inherent ability to manage the trade-off between fairness and performance. This suggests that Fair Random Forest may be particularly suited for scenarios where having control over the balance between these measures is important. In contrast, Fair Logistic Regression and Fair Adversarial Learning showed less sensitivity to changes in their fairness parameters, not achieving the same low unfairness scores as Fair Random Forest. However, pairing these algorithms with fairness-aware hyperparameter optimisation led to the desired, previously mentioned control and to much lower unfairness scores. While ultimately the best choice depends on preference, in terms of fairness combining fair machine learning algorithms with fairness-aware hyperparameter optimisation shows promising results.

For the ILT dataset, depending on the desired balance between predictive performance and fairness, either Fair Random Forest, Fair Logistic Regression paired with fairness-aware hyperparameter optimisation, or Fair Adversarial Learning paired with fairness-aware hyperparameter optimisation is the most sensible choice. Looking at the results for the Adult dataset, which were obtained to help assure methodological quality and provide a robust foundation for our experimental design, Fair Random Forest shows the highest performance overall, the lowest unfairness overall, and the most reliable control over the performance-fairness trade-off.

8.1 Limitations

While this work provides valuable insights, it has some limitations that must be acknowledged. First, the ILT dataset classification task is difficult to solve, given the heterogeneity (non-uniformity) and sparsity of the target-to-sensitive attribute distribution. This is why we conducted our experiments on one other dataset. While our experiments were thoroughly performed and our results are valid, additional experimentation on benchmark datasets would increase the robustness of our work.

We have conducted our experiments using Random Forest, Logistic Regression, Neural Network, Fair Random Forest, Fair Logistic Regression, and Fair Adversarial Learning. There exist many other (fair) machine learning algo-

rithms for binary classification that we have not experimented with. Hence, caution should be taken with generalising the findings to other algorithms. For hyperparameter optimisation, we used the Tree-structured Parzen Estimator algorithm. Fairness-aware hyperparameter optimisation can be used in combination with any hyperparameter optimisation algorithm. We have not experimented with other algorithms, and our findings may not be generalisable to them.

The maximum number of evaluations for Hyperopt was set to 100 to be able to realise a reasonable computation budget. Setting a higher maximum number of evaluations could potentially affect results. Similarly, adjusting the ranges that we set for the hyperparameters (found in Appendix B) might affect results.

8.2 Future Work

Given the current results, several directions appear promising for future work. Specifically for the ILT dataset, incorporating more sophisticated techniques for dealing with small and/or imbalanced datasets, such as advanced sampling methods, cost-sensitive learning, outlier detection or model combination, could potentially enhance the performance and fairness of algorithms (Ramyachitra and Manikandan, 2014).

Extending the experiments to multi-class classification tasks and other tasks that involve sensitive attributes could provide more insights into the flexibility and robustness of combining in-processing fairness algorithms with fairness-aware hyperparameter optimisation. As of now, we only experimented with binary classification tasks. It would be interesting to find out if our findings are generalisable to other tasks.

Exploring alternative fair classification algorithms, hyperparameter optimisation algorithms, and (threshold-independent) fairness measures and their impact on predictive performance across a range of datasets would also be beneficial. We conducted our experiments with three traditional classification algorithms and three fair "counterpart" classification algorithms. Our findings may be tested for robustness by conducting the experiments with different traditional and fair classification algorithms. Similarly, we conducted our experiments with one hyperparameter optimisation algorithm and we would be interested if the results are affected by utilising different hyperparameter optimisation techniques. As a performance measure, we used the ROC-AUC score. As a fairness measure, we used the strong demographic parity score. Using different threshold-independent performance measures such as log-loss and/or different threshold-independent fairness measures such as matching conditional frequencies (Hardt et al., 2016) could provide more insights into the flexibility of combining in-processing fairness algorithms with fairness-aware hyperparameter optimisation.

Lastly, an analysis of the trade-offs between computational cost, performance, and fairness could aid practitioners in their decision making.

9 Conclusion

This work explored the application of fair machine learning algorithms on the binary classification task related to predicting illegal ship breaking for the ILT and the intricate balance between performance and fairness. Three traditional machine learning algorithms were implemented: (1) Random Forest, (2) Logistic Regression, and (3) Neural Network, and three fair machine learning algorithms were implemented: (1) Fair Random Forest, (2) Fair Logistic Regression, and (3) Fair Adversarial Learning. We experimented with the traditional and fair machine learning algorithms optimised for classification performance and the traditional and fair machine learning algorithms optimised for both classification performance and fairness simultaneously.

For the traditional machine learning algorithms, our findings show that Random Forest had the highest predictive performance scores, but also showed the highest unfairness scores. Logistic Regression had the lowest predictive performance scores, but also showed the lowest unfairness scores. Neural Network scored in between these two on both counts. We found that employing traditional machine learning paired with fairness-aware hyperparameter optimisation greatly improves fairness over "plain" traditional machine learning. For the fair machine learning algorithms, our findings suggest that Fair Random Forest offers the most reliable solution for balancing performance and fairness in comparison to the two remaining fair algorithms. Its fairness parameter Θ offers control over the performance-fairness trade-off. For Fair Logistic Regression, its fairness parameter *constraint weight* had a much smaller impact on both performance and fairness. For Fair Adversarial Learning, adjusting its fairness parameter α had more influence on fairness than on performance on both datasets. For the latter two algorithms, while ultimately their performance-fairness trade-off points lacked with respect to the Fair Random Forest, their tunability was severely improved by enhancing the algorithms with fairness-aware hyperparameter optimisation by incorporating a compound performance-fairness objective function into the hyperparameter optimisation step. When inspecting the results on the Adult dataset, Fair Random Forest showed both the highest ROC-AUC, the lowest strong demographic parity, the largest performance-fairness trade-off range, and the most reliable control over this trade-off. Ultimately, for the ILT dataset, either Fair Random Forest, Fair Logistic Regression paired with fairness-aware hyperparameter optimisation, or Fair Adversarial Learning paired with fairness-aware hyperparameter optimisation is the best choice. This depends on the desired balance between classification performance and fairness.

In conclusion, we presented the following contributions: (1) we gave recommendations for the improvement of the fairness of the ILT's ship breaking model, (2) we proposed a functional model-agnostic fairness-aware objective function, based on SCAFF in Fair Random Forest, and (3) we proposed the practice of employing fair machine learning algorithms in combination with fairness-aware hyperparameter optimisation. This work advances our knowledge in the field of fair machine learning and provides a foundation for future work.

Bibliography

Agarwal, A., Beygelzimer, A., Dudík, M., Langford, J., and Wallach, H. M. (2018). A Reductions Approach to Fair Classification. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 60–69. PMLR.

Barata, A. P., Takes, F. W., van den Herik, H. J., and Veenman, C. J. (2024). Fair tree classifier using strong demographic parity. *Machine Learning*, 113(5):3305–3324.

Becker, B. and Kohavi, R. (1996). Adult. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5XW20.

Bellamy, R. K. E., Dey, K., Hind, M., Hoffman, S. C., Houde, S., Kannan, K., Lohia, P., Martino, J., Mehta, S., Mojsilovic, A., Nagar, S., Ramamurthy, K. N., Richards, J. T., Saha, D., Sattigeri, P., Singh, M., Varshney, K. R., and Zhang, Y. (2018). AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias. *Computing Research Repository*, abs/1810.01943.

Bergstra, J., Yamins, D., and Cox, D. D. (2013). Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 115–123. JMLR.org.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1):5–32.

Caton, S. and Haas, C. (2024). Fairness in Machine Learning: A Survey. *ACM Computing Surveys*, 56(7):166:1–166:38.

Cruz, A. F., Saleiro, P., Belém, C., Soares, C., and Bizarro, P. (2021). Promoting Fairness through Hyperparameter Optimization. In *IEEE International Conference on Data Mining, ICDM 2021, Auckland, New Zealand, December 7-10, 2021*, pages 1036–1041. IEEE.

de Bruin, G. J., Barata, A. P., van den Herik, H. J., Takes, F. W., and Veenman, C. J. (2022). Fair automated assessment of noncompliance in cargo ship networks. *EPJ Data Science*, 11(1):13.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. S. (2011). Fairness Through Awareness. *Computing Research Repository*, abs/1104.3913.

European Commision (2024). AI Act. `https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai`. Last accessed on 12-04-2024.

Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., and Venkatasubramanian, S. (2015). Certifying and Removing Disparate Impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, pages 259–268. ACM.

Goodman, R. (2018). Why Amazon's automated hiring tool discriminated against women. *American Civil Liberties Union*.

Grant, C. (2022). Algorithms Are Making Decisions About Health Care, Which May Only Worsen Medical Racism. *American Civil Liberties Union*.

Hardt, M., Price, E., and Srebro, N. (2016). Equality of Opportunity in Supervised Learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3315–3323.

Hinton, G. E. (1989). Connectionist Learning Procedures. *Artificial Intelligence*, 40(1-3):185–234.

Jiang, R., Pacchiano, A., Stepleton, T., Jiang, H., and Chiappa, S. (2019). Wasserstein Fair Classification. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, volume 115 of *Proceedings of Machine Learning Research*, pages 862–872. AUAI Press.

Kamishima, T., Akaho, S., Asoh, H., and Sakuma, J. (2012). Fairness-Aware Classifier with Prejudice Remover Regularizer. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II*, volume 7524 of *Lecture Notes in Computer Science*, pages 35–50. Springer.

Larson, J., Mattu, S., Kirchner, L., and Angwin, J. (2016). How We Analyzed the COMPAS Recidivism Algorithm. *ProPublica*.

Lloyd's List intelligence (2024). The complete view of Maritime Data. `https://www.lloydslistintelligence.com/about-us/our-data`. Last accessed on 12-04-2024.

Lu, Z. and Yin, M. (2021). Human Reliance on Machine Learning Models When Performance Feedback is Limited: Heuristics and Risks. In *CHI '21: CHI Conference on Human Factors in Computing Systems, Virtual Event / Yokohama, Japan, May 8-13, 2021*, pages 78:1–78:16. ACM.

Mason, S. J. and Graham, N. E. (2002). Areas beneath the relative operating characteristics (ROC) and relative operating levels (ROL) curves: Statistical significance and interpretation. *Quarterly Journal of the Royal Meteorological Society*, 128(584):2145–2166.

McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models*. Springer.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. (2022). A Survey on Bias and Fairness in Machine Learning. *ACM Computing Surveys*, 54(6):115:1–115:35.

Ministerie van Infrastructuur en Milieu (2023). Home - Inspectie Leefomgeving en Transport (ILT). `https://english.ilent.nl/`. Last accessed on 12-04-2024.

NGO Shipbreaking Platform (2024). NGO Shipbreaking platform. `https://shipbreakingplatform.org/`. Last accessed on 12-04-2024.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., VanderPlas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Petersen, F., Mukherjee, D., Sun, Y., and Yurochkin, M. (2021). Post-processing for individual fairness. *Advances in Neural Information Processing Systems*, 34:25944–25955.

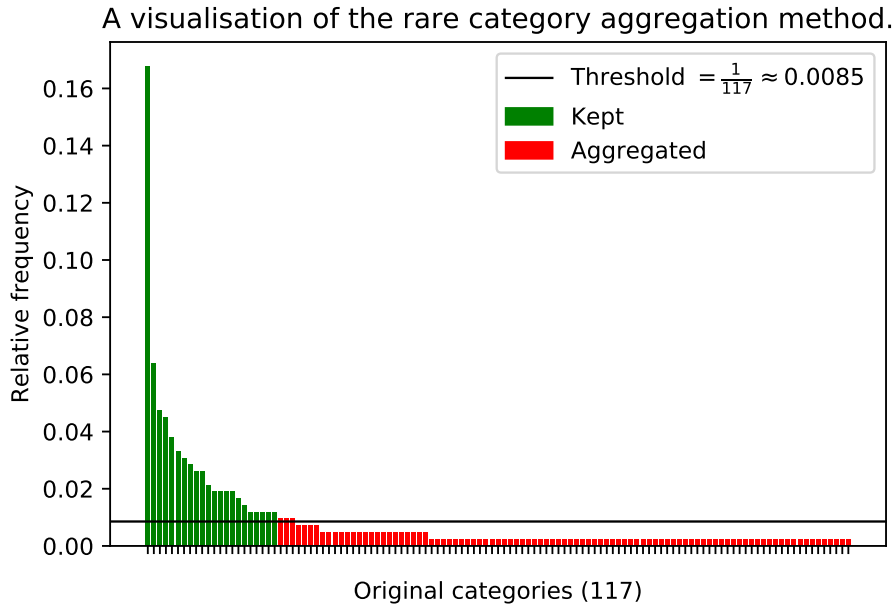
Ramyachitra, D. and Manikandan, P. (2014). Imbalanced dataset classification and solutions: a review. *International Journal of Computing and Business Research*, 5(4):1–29.

Weerts, H. J. P., Pfisterer, F., Feurer, M., Eggensperger, K., Bergman, E., Awad, N. H., Vanschoren, J., Pechenizkiy, M., Bischl, B., and Hutter, F. (2023). Can Fairness be Automated? Guidelines and Opportunities for Fairness-aware AutoML. *Computing Research Repository*, abs/2303.08485.

Zhang, B. H., Lemoine, B., and Mitchell, M. (2018). Mitigating Unwanted Biases with Adversarial Learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, AIES 2018, New Orleans, LA, USA, February 02-03, 2018*, pages 335–340. ACM.

A Clamper

Figure 9: The rare category aggregation method for an example use case with 117 original categories and 23 resulting categories.



B Hyperparameters

Table 3: The hyperparameters that were optimised for each algorithm and their ranges.

Algorithm	**Model**	**Hyperparameters**
Random Forest classifier	RandomForestClassifier	$n\_estimators \in [10 \,..\, 1,000]$, $criterion \in \{\text{'}gini\text{'}, \text{'}entropy\text{'}\}$, $max\_depth \in \{None, [2 \,..\, 200]\}$, $min\_samples\_split \in [2 \,..\, 10]$, $min\_samples\_leaf \in [1 \,..\, 10]$, $max\_features \in \{\text{'}sqrt\text{'}, \text{'}log2\text{'}, None\}$, $bootstrap \in \{True, False\}$, $class\_weight$ $\in \{\text{'}balanced\text{'}, \text{'}balanced\_subsample\text{'}, None\}$
Logistic Regression classifier	LogisticRegression	$penalty \in \{\text{'}l1\text{'}, \text{'}l2\text{'}, \text{'}elasticnet\text{'}, None\}$, $tol \in [0.00001, 0.001]$, $C \in [0.01, 10]$, $fit\_intercept \in \{True, False\}$, $class\_weight \in \{\text{'}balanced\text{'}, None\}$, $max\_iter \in [10 \,..\, 1,000]$, $l1\_ratio \in [0, 1]$
Neural Network classifier	AdversarialDebiasing (debias=False)	$num\_epochs \in [5 \,..\, 500]$, $batch\_size \in [8 \,..\, 2,048]$, $classifier\_num\_hidden\_units \in [20 \,..\, 2,000]$
Fair Random Forest	FairRandomForestClassifier	$(theta \in [0, 1])$, $n\_bins = 256$, $bootstrap = True$, $max\_depth \in [1 \,..\, 20]$, $max\_features \in [0.05, 0.95]$, $n\_estimators \in [100 \,..\, 500]$, $min\_samples\_leaf \in [1 \,..\, 10]$, $min\_samples\_split \in [2 \,..\, 10]$
Fair Logistic Regression	GridSearchReduction	$penalty = \text{'}elasticnet\text{'}$, $(constraint\_weight \in [0, 1])$, $grid\_size \in [2 \,..\, 50]$, $grid\_limit \in [0.4, 10]$, $tol \in [0.00001, 0.001]$, $C \in [0.01, 10]$, $l1\_ratio \in [0, 1]$
Fair Adversarial Learning	AdversarialDebiasing (debias=True)	$(adversary\_loss\_weight \in [0, 1])$, $num\_epochs \in [50 \,..\, 500]$, $batch\_size \in [16 \,..\, 1,024]$, $classifier\_num\_hidden\_units \in [40 \,..\, 1,000]$