Analysing the Impact of Data Distribution Shifts on Model Fairness in Machine Learning

(discussion paper)

Federico Motta[1,*], Yijie Li[2], Huiping Chen[2], Federica Mandreoli[1] and Paolo Missier[2]

[1]Department of Physics, Informatics and Mathematics, University of Modena and Reggio Emilia, Via Campi 213/B, 41125, Modena, Italy

[2]School of Computer Science, University of Birmingham, Edgbaston, Birmingham, B15 2TT, UK

Abstract

As decision-making processes are increasingly automated by the deployment of machine learning techniques, being able to *a priori* ensure the fairness of these models has become a concern of paramount importance. This is especially true in high-stake domains, where data are often provided as they are, i.e., without any additional insight from domain experts about the presence of possible sensitive attributes, such as gender, nationality, or religion. At the same time, data distribution may evolve over time, i.e., lead to drifts, potentially harmful for the performance of the deployed models. Thus, the interplay between: (i) proactively monitoring for degradation in accuracy and promptly retraining the models, as well as (ii) being able to grant fairness regardless of the possible bias within the data has further entangled this already tricky challenge. In this paper, we present and analyse a synthetically generated example of data distribution shift affecting the model performance, including its fairness. Then, we show how only focusing on singularly addressing either the accuracy drop, rather than the introduced bias, cannot completely solve the issues. In conclusion, we hint at the need for a holistic approach to mitigate both the problems as a possible research direction in this field.

Keywords

Data Drift, Distribution Shifts, Model Performance, Fairness

1. Introduction

The growing spread of Machine Learning (ML) models across different sectors, is having a huge impact on people's lives. However, the goodness of these models is tightly related to the quality of the data used to train them, thus, producing just an accurate model may not always be enough. For example, whenever operating in scenarios where the data contain sensitive attributes, such as healthcare [1] or lending [2], it is of paramount importance to also ensure that the produced models are fair [3, 4], i.e., do not discriminate or perpetuate any form of bias which may be present in the data.

At the same time, real-world data rarely are static and the characteristics of the given input data may change over time, due to natural changes in the observed/sampled population, changes in the data collection procedures [5] or in the treatment protocols of a disease [6]. Thus, many ML models once built for a specific task are then deployed and fed with new data, potentially different from the samples seen during the training phase. These so-called data shifts, if unnoticed can generate concept drifts and degrade the model performance, both in terms of accuracy and fairness. Therefore, taking measures to detect them is crucial to preserve the models fair and accurate even in dynamic scenarios.

In this paper we provide background on the key concepts of data shift and model fairness, and then illustrate, through a simple example, the problem of preserving fairness in the presence of data shift, and how fairness and performance stand in contrast to each other in such dynamic scenarios.

SEBD 2025: 33rd Symposium On Advanced Database Systems, June 16–19, 2025, Ischia, Italy

*Corresponding author.

federico.motta@unimore.it (F. Motta); yxl2056@student.bham.ac.uk (Y. Li); h.chen.13@bham.ac.uk (H. Chen); federica.mandreoli@unimore.it (F. Mandreoli); p.missier@bham.ac.uk (P. Missier)

0000-0002-5946-0154 (F. Motta); 0009-0005-1008-6666 (Y. Li); 0000-0003-1782-667X (H. Chen); 0000-0002-8043-8787 (F. Mandreoli); 0000-0002-0978-2446 (P. Missier)

2. Background and related work

The training of machine learning models relies on the assumption that input data are independent and identically distributed (i.i.d.). However, the dynamism of real-world environments can cause unexpected changes in data distributions, which might lead to a degradation of the model performance or a mutation in the fairness of the ML-model outputs. Moreover, the absence of a clear and unified terminology, about the various types of changes in the input data, further entangles this challenge; especially in the already intricate Big Data scenario where data distributions are often unknown.

2.1. Data distribution shifts

Given a target variable and a model attempting to predict it, e.g., over time, the term *concept drift* refers to any alteration of the statistical properties of that variable [7]. Different types of concept drifts exist [8]: *covariate shifts* [9] concern changes in distribution between the training data and the test data, while maintaining the conditional probability distribution of the target given the input; depending on the effect this has on the target variable, we have *virtual* and *real drifts* [10], where the former don't affect the concept, whilst the latter do. *Prior-probability shifts* (sometimes referred to as *label* or *target shifts*) [11, 12, 13] are the opposite of covariate shifts, namely the conditional probability distributions of the outcome given the input are coherent across the training and testing phases, but the target probability distributions change. Moreover, depending on the extent of the change in the data distribution, we may have *local drifts* [10], i.e., only occurring in some regions of the instance space and not at a dataset level; or just the addition of new attributes/target classes, such as in the case of *feature/concept evolution* [14] or their disappearance (*concept deletion*) [15].

Time-wise, concept drifts are instead classified depending on the pattern followed by their arrival [16]: *sudden drifts* [17] occur when the transition from a concept to another is abrupt in time, *gradual drifts* [18] happen when the target distribution undergoes progressive transformations, *recurring drifts* [19] are like gradual ones but they characterised by a periodic transition which evenly phases in/out the new/old concepts, and finally, *incremental drifts* [20] are about the replacement of the old concept in a slow and continuous manner (with respect to the gradual drift they don't have a clear boundary separating the two concepts, but rather a fading window).

2.1.1. Drift detection methods

Being able to promptly detect concept drifts is crucially important to keep stable machine learning models' performance in dynamic environments where the data distributions of the underlying data evolve over time [16]. Although many *supervised methods* capable of detecting concept drift exist in literature [21], they are often impractical to use because of their reliance on the availability of true labels, requiring instead *unsupervised techniques*. A taxonomy of the possible categories has been proposed to reflect use case scenarios [22]. For streaming data, *online-based approaches* use a reference fixed [23, 24] or sliding [25, 26] window to check for drift at every arriving instance. *Batch-based approaches* accumulate instances and use them all, as in *whole-batch approaches* [27, 28], or only a subset of them, like in *partial-batch approaches* [29, 30] to detect the drift. DriftLens [31] is an unsupervised drift detection method that applies specifically to deep learning models and real-time applications. Its execution consists of an offline phase, where the baseline distributions and thresholds are estimated, and an online phase, where new data streams are analysed using fixed-size windows. Moreover, in each window per-batch and per-label distributions are computed and compared with the baseline; drifts are predicted whenever the distances between the distributions exceed the thresholds.

The majority of the methods mentioned above rely on distances [32] between data distributions to quantify the severity of drifts, including: the Kullback-Leibler (KL) divergence (a measure of relative entropy), the Hellinger distance (a symmetric version of the KL-divergence), the Jensen-Shannon divergence (a smoothed version of the KL-divergence), the Wasserstein-1-distance, the Frechét Inception Distance [31] (between normal distributions), but also techniques intersecting neighbourhoods, or

leveraging statistical tests such as the Maximum Mean Discrepancy, the Kolmogorov-Smirnov test (which allow incremental sampling over time) and Hoeffding's inequality-based bound identification [33] (for independent and bounded random variables).

2.1.2. Approaches to handle drifts

The simplest reaction to drift is to retrain the entire model. This however is inefficient, and more practical methods have been developed [17].

Instance selection This technique selects only relevant instances based on the currently learned target, whilst filtering out irrelevant, noisy or redundant samples. Mostly adopted in online learning contexts, these methods use a sliding window to define a short-term memory and sample relevant instances [10, 27]. However, they are vulnerable to local or recurrent drifts, when the fixed window size is shorter than the drift transition time [34].

Instance weighting This approach leverages the capability of some models, such as Support Vector Machines, to weight instances accordingly with their age or relevance [34]; allowing to gradually shift the focus, as the target evolves, e.g., using age as a proxy for an exponentially decaying memory. At the same time, this technique is unfortunately more prone to over-fitting [10], hinting that more complex instance selection heuristics, like those maintaining the window-size dynamic, usually perform better.

Ensemble learning This method leverages a set of models, built over different time periods. The key to keep the focus on the current actual concept is combining several sub-models' predictions. For example, tuning the low/high diversity of the newly trained ensembles with the respect to the type of detected drift, may outperform models trained from scratch after the drift as already occurred [35]. In [16, 33] is instead extended the instance weighting approach, with simpler (e.g. binary) classifiers specialised on each class. These are later combined with aggregation techniques like majority/weighted voting. However, maintaining a complex learning architecture may not always justify its benefits over less accurate but simpler predictors.

Active learning on harmful data drifts Finally, recent works [36, 7] introduced *harmful* and *benign* data drifts, i.e., the ability to state whether a drift in the input data can cause a concept drift capable of degrading the ML model performance. In [36] ensembles of Constrained Disagreement Classifiers (CDCs) trained to agree on training data and disagree on test data are used to analyse the ratio of disagreement and thus detect harmful data drifts. Whilst in [7] are introduced Data Distributions with Low Accuracy (DDLAs) identifiers, i.e., subsets of the feature space where the accuracy of the model is lower than its overall performance. These, if respectively measured on the training/test data, can determine the harmfulness too, suggesting the need of completely retrain the model or just fine-tune it on a sampling of the misclassified testing instances, once they have been labelled by experts.

2.2. Fairness

A crucial aspect of the application of ML algorithms to real-world data is the ability to ensure their fairness, i.e., grant that their behaviour will not perpetuate prejudice or societal bias against any sub-population of individuals because of their inherent or acquired characteristics [4]. In order to measure the amount of fairness of an algorithm we first need to introduce the concept of *protected or sensitive attribute* [3], i.e., any feature capable of partitioning individuals into groups sharing similar benefits [1]; e.g., a non-exhaustive list of attributes traditionally behaving like this is: disability, gender expression/identity, health/marital status, nationality, race, religion, sex and sexual orientation [37].

2.2.1. Fairness metrics

Multiple formal definitions of algorithmic fairness have been proposed, and taxonomies are emerging [38, 2, 1, 4]. These definitions are usually built around three fundamental aspects of a classifier: (i) independence, i.e., not taking into account the potential correlation between the prediction and the sensitive attribute, (ii) separation, or the amount of non-correlation between the prediction and the conditional of the sensitive attribute given the target variable and (iii) sufficiency, which aims at keeping independent the target variable and the conditional of the sensitive attribute given the prediction. With these *abstract fairness criteria* in mind, a coarse grain distinction in between *group*, *sub-group* and *individual fairness* definitions can be identified.

Group fairness According to [38], group-based fairness metrics essentially compare the outcomes of a classifier trained to distinguish between two or more groups defined by considering the sensitive attribute. Among these:

- **Parity-based metrics**, compare predicted Positive Rate (PR) between the groups. For instance, *Statistical/Demographic Parity* (DP) [39, 40] ensures that individuals from the protected and non-protected groups are equally probable of having a positive result, i.e.:

$$p(\hat{y} = 1 \mid z_1) = \cdots = p(\hat{y} = 1 \mid z_N) \tag{1}$$

  where $\hat{y}$ is the predictor, and z the sensitive attribute defining N groups, $z_1, \ldots, z_n$. For simplicity, z is often considered Boolean and thus only taking values in $\{0, 1\}$. Inversely, *Disparate Impact* [41] controls the fraction of positive predictions given the sensitive attribute being unset vs those predicting the same outcome but given the sensitive attribute being set;
- **Confusion matrix-based metrics**, leverage instead the true/false positive/negative rates. For example, *Equal Opportunity* (EO) [39, 42] ensures the same chances of positive outcomes for all individuals regardless of the group they belong to, *Equalised Odds* [43, 42] aim at achieving the same rate of true positives and false positives on different groups, *overall accuracy equality* seeks for the same accuracy on each protected group, *conditional use accuracy equality* tries to balance the false omission rate and false discovery rate, *Treatment Equality* (TE) aims at the same ratio of false negative and false positives across the groups [44], *conditional equal opportunity* [45] permits to equally weight opportunities on a given sensitive attribute, *Average Odds Difference* [46] is the average between the False PR and the True PR, *conditional statistical parity* [47] ensures that given a limited set of sensitive attributes, an equal proportion of individuals sharing the same values are detained in each group;
- **Calibration-based metrics**, only consider the predicted probability or score, like the *Well calibration*, or *test fairness/calibration/matching conditional frequencies fairness* [48];
- **Score-based metrics**, finally try to balance the positive and negative classes, like *Bayesian fairness* [49] does.

Individual fairness As opposed to group-based metrics; individual fairness considers the outcome for each participating individual. For instance, *counterfactual fairness* [50] stems from the intuition that a decision is fair if it holds both in the actual world and in a counterfactual one, where the individual belongs to a different group, *contrastive fairness* [51] instead compares the outcomes between similar individuals under all the relevant aspects except for their values on the sensitive attribute, *equality of efforts* [52] focus on which effort should be made in order to achieve the same outcome predicted for individuals having a different value of sensitive attribute. All these three metrics have their roots in causal models, while the *Generalised Entropy Index* (GEI) [53] measure the individual impact on the prediction in a manner similar to the Gini Index [54]. The *Theil Index* is just a special case of GEI, when the parameter $\alpha = 1$. Finally, also *Fairness through Awareness/Unawareness* [55, 56, 50] fall under individual fairness too. In detail, the former states that fixed a similarity metric, similar individuals should receive similar outcomes; whilst the latter defines the fairness of an algorithm as its capability of non-explicitly use sensitive attributes in the decision-making process.

Sub-group fairness Last but not least, sub-group fairness [57, 58] tries harmonising both the group and individual fairness objectives; e.g. by picking a Group Fairness Indicator (GFI) [59, 2] and wondering whether it also holds on to a wider collection of subgroups.

In this paper, we primarily focus on group fairness. Given its better visualisation capability and simplicity, we mainly use DP (as defined in Eq. 1) to measure fairness when illustrating the problem in Sec. 3.

2.3. Fairness-specific distribution shifts

Among all the data distribution shifts described in Sec. 2.1, some influence more than others fairness. According to the centralisation on sensitive attributes or relationship between sensitive attributes and labels, these shifts can be distinguished as three primary categories: *demographic shifts* [60, 61] denote the distribution changes of sensitive variables that are highly associated with fairness; therefore, a model that is fair in the training data may struggle to maintain fairness on the deployment data due to the altered group proportions. *Sub-population shifts* [62] refer to a particular subgroup with specific values of sensitive attributes and labels having fewer positively labelled samples in the training phase and an increased proportion in the deployment phase. Last but not least, *Correlation shifts* [63, 64] contribute to change the dependence relationship between sensitive attributes and labels, proposing a straightforward strategy to address fairness problems in the context of dynamic environments.

3. Problem Illustration

We now present an example to show the effect of data shift on the accuracy of a binary classifier, and on its fairness relative to a single binary sensitive attribute. The example shows that a perfect and fair classifier, trained on a linearly separable dataset, loses both accuracy and fairness in the presence of shift, specifically when there is a shift in correlation between the sensitive attribute (z) and other covariates (x), denoted x-z *correlation shift*. Furthermore, we show that the simple approach of retraining the model to optimise for either of the two objectives degrades the other. This justifies further work into preserving accuracy-fairness trade-offs in the presence of data shift.

3.1. Notation and Problem Setup

Let $\mathbf{x} \in \mathbb{X}$ where $\mathbb{X}$ denotes the feature space and let $y \in \mathbb{Y}$ where $\mathbb{Y}$ denotes the label space. The sensitive attributes are denoted $\{z_i\}_{i=1}^{N}$ (e.g., gender, race), $z_i \in \mathbb{X}$; for simplicity, we focus on the case with a single sensitive attribute z. At time t_0, a model $M(D_0)$ is trained on an original dataset $D_0 = \{(\mathbf{x}_i, z, y_i)\}_{i=1}^{|D_0|}$, where there exist specific correlations among the feature vector $\mathbf{x}$, the sensitive attribute z, and the label y. However, at time t_1, a shift may occur in the data distribution such that the original correlations between the three variables are altered or even broken. This change can adversely affect the performance of $M(D_0)$, impacting both predictive accuracy and fairness properties. In this section, we illustrate a relatively unnoticed circumstance leveraging a classifier f_θ, parameterised by $\theta \in \Theta$, and investigating the shifts in data, namely the disruption of the $\mathbf{x}$–z correlation, which can lead to degradation in model accuracy and fairness.

3.2. Running Example—with x-z Correlation Changes

Consider a model designed to decide whether an individual can be granted a loan. Here $x =$ *income* and $z =$ *age* are the only covariates. This yields the two sensitive groups: the older group (GO), with $\text{age} \geq 40$, and the younger group (GY), with $\text{age} < 40$. In addition, being granted a loan is the positive class ($y = 1$), and being rejected is the negative class ($y = 0$). Sec. 3.2.1 describes the settings of our experiments in detail. The results and analysis are shown in Sec. 3.2.2.

3.2.1. Simulation Settings

To characterise and evaluate the impact on the model performance of shifts in the data distribution, we design an experiment using synthetic data. This synthetic dataset consists of 2,000 records and comprises two continuous and normally distributed features: *income* and *age* independently generated, and the loan approval, a binary variable used as classification target. More specifically:

1. *income* follows a bimodal distribution, *high*-income cases are generated to be normally distributed around 70, whilst *low*-income ones are centred on 30; both have a standard deviation (std. dev.) of 20. The resulting average ($\overline{\text{income}} = 50$) is equally distant from the two peaks. This feature will not be affected by the shift, i.e., it is the same in both D_0 and D_1.
2. in D_0 the sensitive group attribute *age* is normally distributed around 40 ($\mu_{t_0}^{\text{age}}$) with std. dev. 10 in the $[18, 80]$ range, i.e., it does not depend on the income. In D_1 a shift is instead introduced to reflect a new negative correlation between *age* and *income*; in detail a Gaussian with the same std. dev. but $\mu_{t_1}^{\text{age}} = \mu_{t_0}^{\text{age}} - \text{factor} \cdot (\text{income} - \overline{\text{income}})$ is used. In our experiments factor was set to 0.2. The resulting shift reflects a potential systemic change in the population structure.
3. in order to have *target* labels aligned with the shift, we sampled them with the following sigmoid probabilities: $p_0 = \sigma(\text{income} - \overline{\text{income}})$ before the shift, and $p_1 = \sigma((\text{income} - \overline{\text{income}}) - (\text{age} - \mu_{t_0}^{\text{age}}))$ after. Hence, if we draw a sample above 0.5 we assign it to Class 1 (C1), i.e., loan approved, or to Class 0 (C0) otherwise. This choice is set to adjust the decision boundary as balance as possible between the different classes, therefore reducing the impact of class imbalance and focusing more on the change in correlation.

This setting creates a realistic scenario, where the correlations between features and the feature–outcome relationships might evolve over time simulating how real-world decision processes might change.

3.2.2. Experimental Results

To ensure rigorous evaluation, we split both the original (D_0) and shifted (D_1) data into $80/20\%$ for the training and test sets. After standardising the data, a 5-fold cross-validation procedure is employed to optimise the hyper-parameters during training, while the test set is exclusively used for the final evaluation. Three models were trained in this experiment: (i) a standard logistic regression classifier M_0 trained on D_0, yielding $\beta_0^{\text{income}} = 9.347$, $\beta_0^{\text{age}} = 0.011$ as weights, and an intercept of 0.255; (ii) a new logistic regression model M_a trained directly on D_1 and optimised for accuracy, with weights $\beta_a^{\text{income}} = 8.428$, $\beta_a^{\text{age}} = -3.451$ and an intercept of 0.247; (iii) a model M_f optimised for fairness, i.e., with a linear decision boundary that only optimises model fairness while ensuring the overall accuracy remains above a threshold. Specifically, M_f is obtained by performing a grid search to optimise a linear classifier of the form $f(\mathbf{x}) = w_1 \cdot \text{income} + w_2 \cdot \text{age} + b$, subject to minimising the absolute difference in positive PR between the two sensitive groups under the constraint that the drop in accuracy is lower than 30%. The optimal parameters obtained are $w_1 = 0.036$, $w_2 = 0.053$, indicating that *age* and *income* are similarly weighted, and $b = -4.090$. Additionally, the original model M_0 is evaluated on both D_0 and D_1 while M_a and M_f are only assessed on D_1.

Fig. 1 depicts the changes in the data vertically, i.e., D_0 in the top panel and D_1 in the bottom ones; while the lines represent how the three models' decision boundaries partition the feature space. Tab. 1 reports the number of data points in each specific sub-group both in the original and shifted datasets. One can observe that in the original test set the older group is slightly favoured; whilst after adding negative correlations between *income* and *age*, the younger group is clearly more favoured.

Before the $\mathbf{x}$–z correlation shift, labels only depend on income because they are generated by a logistic model predicting the probability of an approved loan application. Consequently, as shown in Tab. 2; M_0 since trained on D_0, achieves perfect accuracy (1.000) with low bias (DP $= 1.114$). Therefore, we can also see from the top panel in Fig. 1 that M_0 separates green and blue dots precisely, and the points of the same shape are evenly distributed on both sides of the model. Thus, the original model is both accurate and fair. However, when M_0 is used for inference on shifted data, its accuracy drops to

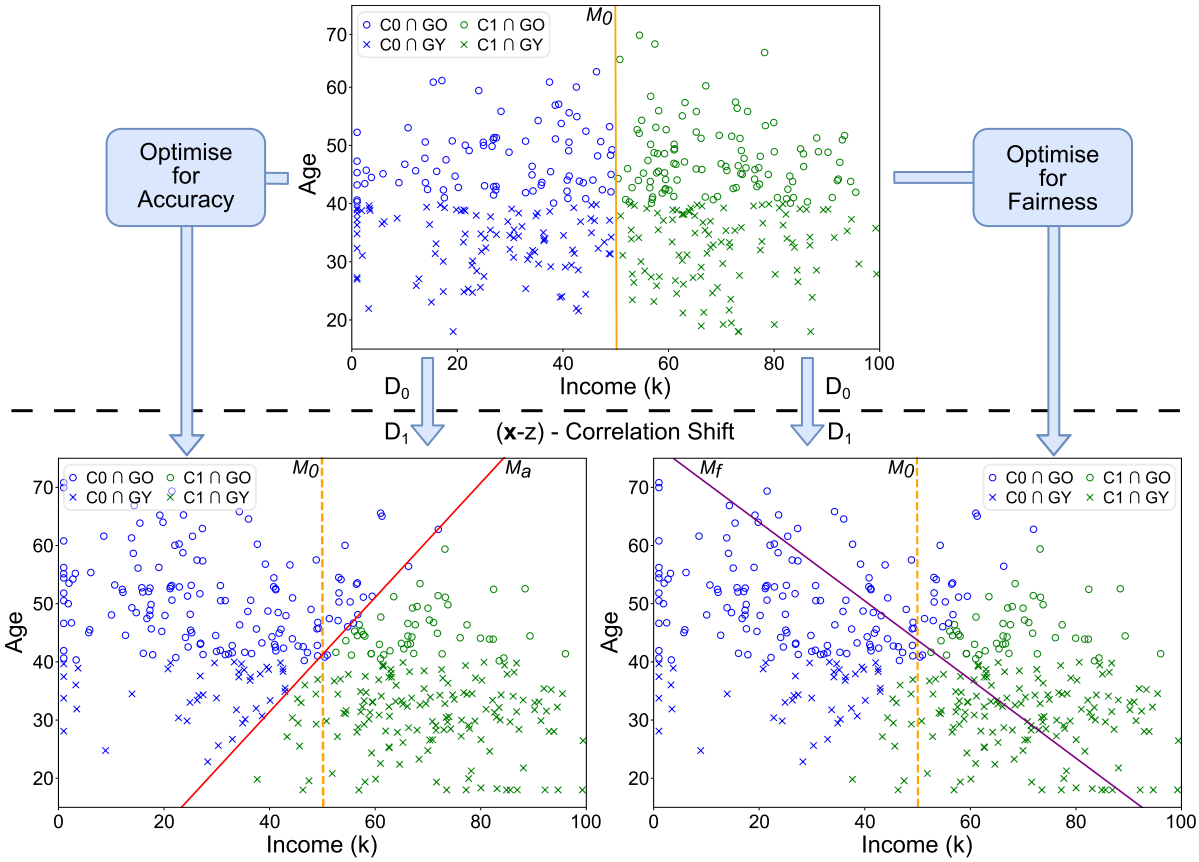

Figure 1: A toy example showing how changing correlations between sensitive attribute *age* and non-sensitive attribute *income* can increase bias. The top panel depicts the data and model before the shift, while the bottom-left/right panels show results from different optimisation strategies: favouring accuracy and fairness, respectively.

Table 1
Per-subgroup data points in original and shifted datasets

Dataset	GO		GY	
	C1	C0	C1	C0
Original	534	463	502	501
Shifted	267	742	772	219

0.915 and its bias to 0.499, clearly favouring the younger group. The shifted data is generated to have the *age* feature negatively correlated with the *income*. In this situation, the positive class is predicted by a logistic model that depends both on *income* and *age*. In the bottom panels in Fig. 1, the shift makes the younger group (GY) being over-represented in the high-income range while the older group (GO) is more concentrated in the low-income range, resulting in imbalanced proportions of approved samples in two groups. The red line in the bottom-left panel of Fig. 1 presents a newly trained model M_a that optimises accuracy, leading to a perfect accuracy of 1.000 (again, because the two classes are linearly separable) but an even more biased DP of 0.322. On the contrary, the purple line representing M_f in the bottom-right panel is optimised for fairness. Thus, this model has nearly zero bias, with a DP of 1.039, whilst still experiencing a significant degradation in terms of accuracy (0.718).

These results confirm that although the initially perfect model M_0 achieves high accuracy and fairness on D_0, when the data distribution changes in a way altering the prior $\mathbf{x}$-z correlations, its performance can degrade. This indicates that a model trained on the old data may suffer from both reduced accuracy and increased bias under the context of data drifts. Meanwhile, the accurate model M_a, since it is not fairness-aware, can have poor fairness on shifted data despite maintaining a high accuracy.

Table 2
Models' performance, in terms of Accuracy, Positive Rate (PR) and *Demographic Parity* (DP) ratio

Test Set	Model	Accuracy	PR(GO)	PR(GY)	DP
Original	M_0	1.000	0.578	0.519	1.114
Shifted	M_0	0.915	0.361	0.723	0.499
	M_a	1.000	0.253	0.786	0.322
	M_f	0.718	0.454	0.437	1.039

4. Concluding remarks

In this paper, we considered the problem of performance degradation in ML models, including both accuracy and fairness, under data distribution shifts and illustrated the problem in a specific scenario. On the one hand, the experiments in Sec. 3 highlight the impact of correlation shifts between sensitive and non-sensitive attributes on model fairness as well as performance, illustrating the challenges of maintaining both high accuracy and fairness in dynamically changing environments. On the other hand, Sec. 3.2.2 demonstrates that focusing solely on accuracy to handle data drifts is not a viable solution for ensuring fairness. In fact, it may even amplify model bias. Instead, explicitly optimising for fairness metrics can effectively mitigate bias caused by distribution shifts—though this comes at a significant cost to accuracy, emphasising the need to carefully balance these trade-offs in real-world applications.

With respect to that, future work will further explore this framework's generalisation capabilities, considering a wider set of models (e.g., tree-based ensembles) and fairness measures (e.g., EO, TE). This comparative analysis will leverage more complex and adherent to real-world scenarios benchmarks [65, 66]. Another interesting research direction will be the study of the robustness to different types of data drift (e.g., sudden, gradual, recurring, etc.) in more dynamic online learning scenarios. Our analysis of the literature and experimental findings clearly indicate the need for methods that not only control fairness during model training but also ensure robustness against distribution shifts during deployment.

Focusing on the data, it is worth noting that real-world data are rarely provided in a way that is readily suitable for the development of ML models [67], thus data engineering pipelines are often used to clean the raw input data through a series of step-by-step transformations resulting in a clean ML-ready datasets. At the same time, it is also known that this data preparation process is often neglected [68]; i.e., once the pre-processing pipeline is built, all the effort is put on the model development and deployment.

On the contrary, we argue that this robustness can be achieved in two ways: (i) proactively monitoring the raw input data going through the data preparation process and alarming the data scientist whenever a shift in the data is detected; or (ii) detecting drifts within the data fed to the ML model and monitoring it for performance degradation. In the former, and less explored case, system functionality is maintained by acting upstream on the data preprocessing pipeline, e.g., updating it to preserve *by design* the fairness of the downstream model by providing it with already balanced and unbiased data. In the latter and more adopted case, system reliability is achieved by repairing the model, e.g., by just fine-tuning it against a bunch of misclassified samples belonging to harmful data drift [7], rather than completely retraining it, as done in Sec. 3.2.2.

One approach does not exclude the other, eventually, minor shifts could be fixed downstream, whilst major accuracy/fairness drops, since usually involving retraining ML models from scratch, could probably benefit from repairs in the data engineering pipeline and in the model design, e.g., by choosing a different trade-off between accuracy and fairness, as suggested by this paper's results.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] R. T. Rabonato, L. Berton, A systematic review of fairness in machine learning, AI and Ethics (2024). doi:10.1007/s43681-024-00577-5.

[2] T. D. Jui, P. Rivas, Fairness issues, current approaches, and challenges in machine learning models, International Journal of Machine Learning and Cybernetics 15 (2024) 3095–3125. doi:10.1007/s13042-023-02083-2.

[3] H. Jagadish, J. Stoyanovich, B. Howe, The Many Facets of Data Equity, Journal of Data and Information Quality 14 (2022) 1–21. doi:10.1145/3533425.

[4] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan, A Survey on Bias and Fairness in Machine Learning, ACM Computing Surveys 54 (2022) 1–35. doi:10.1145/3457607.

[5] K. Zadorozhny, P. Thoral, P. Elbers, G. Cinà, Out-of-Distribution Detection for Medical Applications: Guidelines for Practical Evaluation, in: Multimodal AI in Healthcare: A Paradigm Shift in Health Intelligence, 2023, pp. 137–153. doi:10.1007/978-3-031-14771-5_10.

[6] A. Kore, E. Abbasi Bavil, V. Subasri, M. Abdalla, B. Fine, E. Dolatabadi, M. Abdalla, Empirical data drift detection experiments on real-world medical imaging data, Nature Communications 15 (2024) 1887. doi:10.1038/s41467-024-46142-w.

[7] S. Dong, Q. Wang, S. Sahri, T. Palpanas, D. Srivastava, Efficiently Mitigating the Impact of Data Drift on Machine Learning Pipelines, Proceedings of the VLDB Endowment 17 (2024) 3072–3081. doi:10.14778/3681954.3681984.

[8] F. Bayram, B. S. Ahmed, A. Kassler, From concept drift to model degradation: An overview on performance-aware drift detectors, Knowledge-Based Systems 245 (2022) 108632. doi:10.1016/j.knosys.2022.108632.

[9] M. Sugiyama, M. Kawanabe, Machine learning in non-stationary environments: introduction to covariate shift adaptation, MIT Press, 2012.

[10] A. Tsymbal, M. Pechenizkiy, P. Cunningham, S. Puuronen, Dynamic integration of classifiers for handling concept drift, Information Fusion 9 (2008) 56–68. doi:10.1016/j.inffus.2006.11.002.

[11] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, F. Herrera, A unifying view on dataset shift in classification, Pattern Recognition 45 (2012) 521–530. doi:10.1016/j.patcog.2011.06.019.

[12] Z. Lipton, Y.-X. Wang, A. Smola, Detecting and Correcting for Label Shift with Black Box Predictors, in: Proceedings of the 35th International Conference on Machine Learning, volume 80, 2018, pp. 3122–3130. URL: https://proceedings.mlr.press/v80/lipton18a.html.

[13] K. Zhang, B. Schölkopf, K. Muandet, Z. Wang, Domain Adaptation under Target and Conditional Shift, in: Proceedings of the 30th International Conference on Machine Learning, volume 28, 2013, pp. 819–827. URL: http://proceedings.mlr.press/v28/zhang13d.pdf.

[14] M. M. Masud, Q. Chen, J. Gao, L. Khan, J. Han, B. Thuraisingham, Classification and Novel Class Detection of Data Streams in a Dynamic Feature Space, in: Machine Learning and Knowledge Discovery in Databases, volume 6322, 2010, pp. 337–352. doi:10.1007/978-3-642-15883-4_22.

[15] R. Elwell, R. Polikar, Incremental Learning of Concept Drift in Nonstationary Environments, IEEE Transactions on Neural Networks 22 (2011) 1517–1531. doi:10.1109/TNN.2011.2160459.

[16] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, M. Woźniak, Ensemble learning for data stream analysis: A survey, Information Fusion 37 (2017) 132–156. doi:10.1016/j.inffus.2017.02.004.

[17] A. Tsymbal, The problem of concept drift: definitions and related work, Computer Science Department, Trinity College Dublin 106 (2004).

[18] R. J. Hickey, M. M. Black, Refined Time Stamps for Concept Drift Detection During Mining for Classification Rules, in: Temporal, Spatial, and Spatio-Temporal Data Mining, volume 2007, 2001, pp. 20–30. doi:10.1007/3-540-45244-3_3.

[19] J. B. Gomes, M. M. Gaber, P. A. C. Sousa, E. Menasalvas, Mining Recurring Concepts in a Dynamic

Feature Space, IEEE Transactions on Neural Networks and Learning Systems 25 (2014) 95–110. doi:10.1109/TNNLS.2013.2271915.
[20] R. P. J. C. Bose, W. M. P. Van Der Aalst, I. Žliobaitė, M. Pechenizkiy, Handling Concept Drift in Process Mining, in: Advanced Information Systems Engineering, volume 141, 2011, pp. 391–405. doi:10.1007/978-3-642-21640-4_30.
[21] R. S. M. Barros, S. G. T. C. Santos, A large-scale comparison of concept drift detectors, Information Sciences 451-452 (2018) 348–370. doi:10.1016/j.ins.2018.04.014.
[22] R. N. Gemaque, A. F. J. Costa, R. Giusti, E. M. Dos Santos, An overview of unsupervised drift detection methods, WIREs Data Mining and Knowledge Discovery 10 (2020) e1381. doi:10.1002/widm.1381.
[23] Y. Kim, C. H. Park, An Efficient Concept Drift Detection Method for Streaming Data under Limited Labeling, IEICE Transactions on Information and Systems E100.D (2017) 2537–2546. doi:10.1587/transinf.2017EDP7091.
[24] A. M. Mustafa, G. Ayoade, K. Al-Naami, L. Khan, K. W. Hamlen, B. Thuraisingham, F. Araujo, Unsupervised deep embedding for novel class detection over data stream, in: 2017 IEEE International Conference on Big Data (Big Data), 2017, pp. 1830–1839. doi:10.1109/BigData.2017.8258127.
[25] R. F. De Mello, Y. Vaz, C. H. Grossi, A. Bifet, On learning guarantees to unsupervised concept drift detection on data streams, Expert Systems with Applications 117 (2019) 90–102. doi:10.1016/j.eswa.2018.08.054.
[26] F. Pinagé, E. M. Dos Santos, J. Gama, A drift detection method based on dynamic classifier selection, Data Mining and Knowledge Discovery 34 (2020) 50–74. doi:10.1007/s10618-019-00656-w.
[27] A. G. Maletzke, D. M. Dos Reis, G. E. A. P. A. Batista, Combining instance selection and self-training to improve data stream quantification, Journal of the Brazilian Computer Society 24 (2018) 12. doi:10.1186/s13173-018-0076-0.
[28] B. Li, Y.-j. Wang, D.-s. Yang, Y.-m. Li, X.-k. Ma, FAAD: an unsupervised fast and accurate anomaly detection method for a multi-dimensional sequence over data stream, Frontiers of Information Technology & Electronic Engineering 20 (2019) 388–404. doi:10.1631/FITEE.1800038.
[29] A. F. J. Costa, R. A. S. Albuquerque, E. M. D. Santos, A Drift Detection Method Based on Active Learning, in: 2018 International Joint Conference on Neural Networks (IJCNN), 2018, pp. 1–8. doi:10.1109/IJCNN.2018.8489364.
[30] T. S. Sethi, M. Kantardzic, Handling adversarial concept drift in streaming data, Expert Systems with Applications 97 (2018) 18–40. doi:10.1016/j.eswa.2017.12.022.
[31] S. Greco, B. Vacchetti, D. Apiletti, T. Cerquitelli, Driftlens: A concept drift detection tool, in: Proceedings 27th International Conference on Extending Database Technology, EDBT 2024, Paestum, Italy, March 25 - March 28, 2024, pp. 806–809. doi:10.48786/EDBT.2024.75.
[32] G. I. Webb, R. Hyde, H. Cao, H. L. Nguyen, F. Petitjean, Characterizing concept drift, Data Mining and Knowledge Discovery 30 (2016) 964–994. doi:10.1007/s10618-015-0448-4.
[33] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, G. Zhang, Learning under Concept Drift: A Review, IEEE Transactions on Knowledge and Data Engineering (2018) 1–1. doi:10.1109/TKDE.2018.2876857.
[34] R. Klinkenberg, Learning drifting concepts: Example selection vs. example weighting, Intelligent Data Analysis 8 (2004) 281–300. doi:10.3233/IDA-2004-8305.
[35] L. L. Minku, X. Yao, DDD: A New Ensemble Approach for Dealing with Concept Drift, IEEE Transactions on Knowledge and Data Engineering 24 (2012) 619–633. doi:10.1109/TKDE.2011.58.
[36] T. Ginsberg, Z. Liang, R. G. Krishnan, A Learning Based Hypothesis Test for Harmful Covariate Shift, in: The Eleventh International Conference on Learning Representations, 2023, pp. 1–34. URL: https://openreview.net/forum?id=rdfgqiwz7lZ.
[37] Geneva: Joint United Nations Programme on HIV/AIDS, UNAIDS terminology guidelines, 2024. URL: https://www.unaids.org/en/resources/documents/2024/terminology_guidelines.
[38] S. Caton, C. Haas, Fairness in Machine Learning: A Survey, ACM Computing Surveys 56 (2024) 1–38. doi:10.1145/3616865.

[39] M. Scutari, F. Panero, M. Proissl, Achieving fairness with a simple ridge penalty, Statistics and Computing 32 (2022) 77. doi:10.1007/s11222-022-10143-w.
[40] T. Zhao, E. Dai, K. Shu, S. Wang, Towards Fair Classifiers Without Sensitive Attributes: Exploring Biases in Related Features, in: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, 2022, pp. 1433–1442. doi:10.1145/3488560.3498493.
[41] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, S. Venkatasubramanian, Certifying and Removing Disparate Impact, in: Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 259–268. doi:10.1145/2783258.2783311.
[42] J. Wang, Y. Li, C. Wang, Synthesizing Fair Decision Trees via Iterative Constraint Solving, in: Computer Aided Verification, volume 13372, 2022, pp. 364–385. doi:10.1007/978-3-031-13188-2_18.
[43] V. Perrone, M. Donini, M. B. Zafar, R. Schmucker, K. Kenthapadi, C. Archambeau, Fair Bayesian Optimization, in: Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society, 2021, pp. 854–863. doi:10.1145/3461702.3462629.
[44] R. Berk, H. Heidari, S. Jabbari, M. Kearns, A. Roth, Fairness in Criminal Justice Risk Assessments: The State of the Art, Sociological Methods & Research 50 (2021) 3–44. doi:10.1177/0049124118782533.
[45] A. Beutel, J. Chen, T. Doshi, H. Qian, A. Woodruff, C. Luu, P. Kreitmann, J. Bischof, E. H. Chi, Putting Fairness Principles into Practice: Challenges, Metrics, and Improvements, in: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, 2019, pp. 453–459. doi:10.1145/3306618.3314234.
[46] M. A. U. Alam, AI-Fairness Towards Activity Recognition of Older Adults, in: MobiQuitous 2020 - 17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, 2020, pp. 108–117. doi:10.1145/3448891.3448943.
[47] S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, A. Huq, Algorithmic Decision Making and the Cost of Fairness, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 797–806. doi:10.1145/3097983.3098095.
[48] A. Chouldechova, Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments, Big Data 5 (2017) 153–163. doi:10.1089/big.2016.0047.
[49] C. Dimitrakakis, Y. Liu, D. C. Parkes, G. Radanovic, Bayesian Fairness, Proceedings of the AAAI Conference on Artificial Intelligence 33 (2019) 509–516. doi:10.1609/aaai.v33i01.3301509.
[50] M. J. Kusner, J. Loftus, C. Russell, R. Silva, Counterfactual Fairness, in: Advances in Neural Information Processing Systems, volume 30, 2017, pp. 4066–4076. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/a486cd07e4ac3d270571622f4f316ec5-Paper.pdf.
[51] T. Chakraborti, A. Patra, J. A. Noble, Contrastive Fairness in Machine Learning, IEEE Letters of the Computer Society 3 (2020) 38–41. doi:10.1109/LOCS.2020.3007845.
[52] W. Huang, Y. Wu, L. Zhang, X. Wu, Fairness through Equality of Effort, in: Companion Proceedings of the Web Conference 2020, 2020, pp. 743–751. doi:10.1145/3366424.3383558.
[53] T. Speicher, H. Heidari, N. Grgic-Hlaca, K. P. Gummadi, A. Singla, A. Weller, M. B. Zafar, A Unified Approach to Quantifying Algorithmic Unfairness: Measuring Individual &Group Unfairness via Inequality Indices, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 2239–2248. doi:10.1145/3219819.3220046.
[54] C. Gini, On the measure of concentration with special reference to income and statistics, Colorado College Publication General series 208 (1936).
[55] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, R. Zemel, Fairness through awareness, in: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, 2012, pp. 214–226. doi:10.1145/2090236.2090255.
[56] N. Grgic-Hlaca, M. B. Zafar, K. P. Gummadi, A. Weller, The case for process fairness in learning: Feature selection for fair decision making, in: Symposium on Machine Learning and the Law at the 29th Conference on Neural Information Processing Systems (NIPS 2016), 2016, pp. 45–55. URL: https://www.mlandthelaw.org/papers/grgic.pdf.
[57] M. Kearns, S. Neel, A. Roth, Z. S. Wu, Preventing Fairness Gerrymandering: Auditing and Learning

for Subgroup Fairness, in: Proceedings of the 35th International Conference on Machine Learning, volume 80, 2018, pp. 2564–2572. URL: https://proceedings.mlr.press/v80/kearns18a.html.
[58] M. Kearns, S. Neel, A. Roth, Z. S. Wu, An Empirical Study of Rich Subgroup Fairness for Machine Learning, in: Proceedings of the Conference on Fairness, Accountability, and Transparency, 2019, pp. 100–109. doi:10.1145/3287560.3287592.
[59] A. Torralba, A. A. Efros, Unbiased look at dataset bias, in: CVPR 2011, 2011, pp. 1521–1528. doi:10.1109/CVPR.2011.5995347.
[60] B. An, Z. Che, M. Ding, F. Huang, Transferring Fairness under Distribution Shifts via Fair Consistency Regularization, in: Advances in Neural Information Processing Systems, volume 35, 2022, pp. 32582–32597. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/d1dbaabf454a479ca86309e66592c7f6-Paper-Conference.pdf.
[61] S. Giguere, B. Metevier, Y. Brun, B. C. da Silva, P. S. Thomas, S. Niekum, Fairness Guarantees under Demographic Shift, Proceedings of the 10th International Conference on Learning Representations (ICLR) (2022) 1–24. URL: https://par.nsf.gov/biblio/10334581.
[62] S. Maity, D. Mukherjee, M. Yurochkin, Y. Sun, Does enforcing fairness mitigate biases caused by subpopulation shift?, in: Advances in Neural Information Processing Systems, volume 34, 2021, pp. 25773–25784. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/d800149d2f947ad4d64f34668f8b20f6-Paper.pdf.
[63] Y. Roh, K. Lee, S. E. Whang, C. Suh, Improving Fair Training under Correlation Shifts, in: Proceedings of the 40th International Conference on Machine Learning, volume 202, 2023, pp. 29179–29209. URL: https://proceedings.mlr.press/v202/roh23a.html.
[64] C. Zhao, F. Mi, X. Wu, K. Jiang, L. Khan, C. Grant, F. Chen, Towards Fair Disentangled Online Learning for Changing Environments, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 3480–3491. doi:10.1145/3580305.3599523.
[65] H. Hofmann, Statlog (German Credit Data), UCI Machine Learning Repository, 1994. doi:https://doi.org/10.24432/C5NC77.
[66] Agency for Healthcare Research and Quality (AHRQ), Medical Expenditure Panel Survey (MEPS), 1996. URL: https://meps.ahrq.gov/mepsweb.
[67] F. Mandreoli, D. Ferrari, V. Guidetti, F. Motta, P. Missier, Real-world data mining meets clinical practice: Research challenges and perspective, Frontiers in Big Data 5 (2022). doi:10.3389/fdata.2022.1021621.
[68] N. Sambasivan, S. Kapania, H. Highfill, D. Akrong, P. Paritosh, L. M. Aroyo, "Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI, in: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, 2021, pp. 1–15. doi:10.1145/3411764.3445518.