ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

Drift-Aware Evaluation of Fair Stream Learning

Kathrin Lammers, Fabian Hinder, Barbara Hammer, and Valerie Vaquet *

Bielefeld University, Technical Faculty
Universitätsstraße 25, 33615 Bielefeld, Germany

Abstract. Algorithmic fairness, a key concern in algorithmic decision making, is a well-studied topic in the batch setup. Recently, several extensions for improving fairness in classification tasks have been proposed for the important scenario of non-stationary data streams. Yet, the question of how to reliably evaluate fairness for non-stationary data streams is still open, as popular batch measures can lead to misleading results. Specifically, typically cumulative fairness measures can be problematic if concept drift results in significant changes in model fairness across the data stream. In this contribution, we propose novel fairness scores that are suitable for the streaming scenario, and we demonstrate their suitability on streaming data benchmarks.

1 Introduction

As machine learning algorithms gain increasing importance in people's lives, legislation such as the EU AI Act demands fairness and transparency for these applications. In addition to the development of fair algorithms, the evaluation of model fairness becomes of increasing concern [1]. While the majority of fair learning methods have been developed for the batch setup, there has been a surge in fair online-learning algorithms being developed in recent years as the demand for change-adaptive models increases. But suitable evaluation of these algorithms has so far been neglected [2, 3].

The current state-of-the-art evaluation schemes for computing fairness scores is cumulative evaluation—computing batch fairness metrics over the entire stream [2, 3]. This practice is insufficient for a comprehensive fairness evaluation as it does not account for the effects of concept drift and model adaptation on the fairness scores. In particular, drift might affect which groups are discriminated against over the course of the stream. In severe cases, a cumulative fairness score will cancel these effects, while discrimination is present for most of the stream.

In this work, we propose drift-aware fairness evaluation schemes, which enable a fine-grained analysis of possible biases over the course of time, and we evaluate the results on streaming benchmarks including drift. Unlike current cumulative evaluation measures, we can identify time windows with insufficient group fairness w.r.t. varying groups. After briefly covering the foundations of stream learning and algorithmic fairness (Section 2), we focus on analyzing the pitfalls of state-of-the-art fairness evaluation on data streams (Section 3). We then propose novel evaluation schemes and analyze those in Section 4. Finally, we conclude this work in Section 5.

*Funding in the scope of the BMFTR project KI Akademie OWL under grant agreement No. 16IS24057A and the ERC Synergy Grant "Water-Futures" No. 951424 is gratefully acknowledged.

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

2 Foundations

2.1 Stream Learning

Learning from streaming data deals with a possibly infinite stream of data points $(x_t, y_t), t = 1, 2, \ldots$, where (x_t, y_t) is drawn independently from a time-dependent distribution D_t for timepoint t. Drift refers to the fact that the distribution changes, i.e. $D_{t_0} \neq D_{t_1}$ for some time points $t_0 \neq t_1$. Due to its high relevance for model personalization and lifelong learning, several learning algorithms have been proposed that enable model adaptation in the presence of drift [4, 5]. As for the batch setup, algorithms learning from data streams might lead to violations of algorithmic fairness, especially for imbalanced streams, and a couple of learning methods have been proposed to address this [2, 3].

2.2 Fairness

In human decision-making, fairness is a key concept, referring to the equitable treatment of people while considering their unique circumstances. However, translating it to machine learning is somewhat difficult as multiple, contradictory fairness notions exist, and it may be difficult to decide which are most appropriate for a given context. In this paper, we will focus on the common notion of *group fairness*, which stipulates that different groups, separated by a sensitive attribute S (e.g., race or gender), should be treated equally. In particular, we differentiate between *equality of outcome* and *equality of odds*, which can typically not be fulfilled at the same time. The former requires equal acceptance rates for both groups, with *demographic parity* comparing the absolute difference between groups: $|P[\hat{y}_t = 1|S = 1] - P[\hat{y}_t = 1|S \neq 1]|$, while *disparate impact* uses fractions to compare the acceptance rates: $1 - \frac{P[\hat{y}_t=1|S\neq 1]}{P[\hat{y}_t=1|S=1]}$. Meanwhile, *equality of odds* is composed of the absolute differences in *true positive rates* $|P[\hat{y}_t = 1|S = 1, y_t = 1] - P[\hat{y}_t = 1|S \neq 1, y_t = 1]|$ (also referred to as *equal opportunity*) on the one hand, and the difference of the *false positive rates* $|P[\hat{y}_t = 1|S = 1, y_t = 0] - P[\hat{y}_t = 1|S \neq 1, y_t = 0]|$ on the other hand [1]. In general, a fairness notion F considers a distribution P over datapoints $(x, y) \in (X, Y)$, sensitive attribute $S \subset X$, and predicted label $\hat{y} = f(x)$.

3 Pitfalls of cumulative fairness scoring

In the stream setup, these scores are usually computed prequentially over the entire stream, resulting in a cumulative score over its entire length [3]. Applying these fairness scores, developed for the batch setup, naively to the stream learning setting can yield undesirable results, as the effects of drift can lead to obscured discrimination in the final score. For example, if only the final score over the entire stream is given, it is impossible to say if the classifier first discriminated against one group and then the other, cancelling the previous effect, or if it was consistently free of discrimination.

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

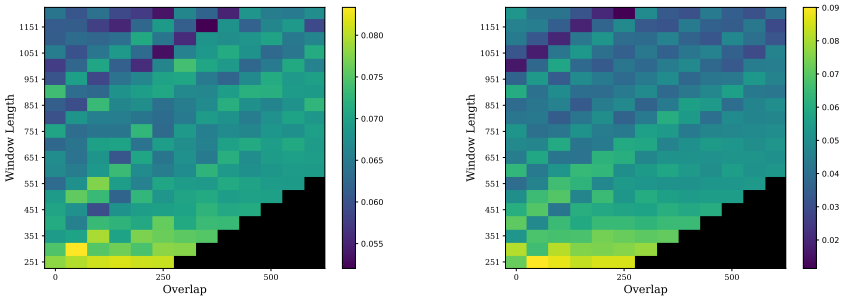

Fig. 1: The effect of window length and overlap on the statistical parity (l.) and equal opportunity (r.) score of classifier FABBOO (EQOP-optimized) on a modified version of the *Student* dataset (grade category change, see [6]). Both hyperparameters have a considerable effect on the overall score.

4 Drift-aware fairness evaluation

To evaluate fairness in a drift-aware manner, there is clearly a need for an overall fairness score that accumulates discrimination over the entire stream in a drift-aware manner. More formally, a drift-aware fairness measure F for the entire stream of length T should average over the time-specific fairness scores F_t: $F = \sum_t \frac{F_t}{T}$ to ensure fairness of all parts of the stream and therefore the stream in its entirety. However, in practice, F is approximated through relative frequency, which is also how we need to compute each F_t. The challenge, therefore, becomes splitting the stream in a manner that allows us to estimate F_t from the resulting segments. For this, the segments should be sufficiently long to give a good estimate of F_t, and free of drift to avoid cumulative effects within the segment.

4.1 Window-based approach

Naively, we can simply split the data stream into multiple windows of equal length, compute the fairness score for each window, and average them at the end of the stream. However, this simple strategy relies on hyperparameters, such as window length and window overlap. This might be an issue as these hyperparameter choices could considerably affect the resulting score.

As can be seen in Fig. 1, if the windows are chosen too small, differences of a few samples, which become increasingly important in streams with high levels of class imbalance, can have drastic effects and lead to proclamations of unfairness where there is—statistically—none. This is because if the windows are too short, P_t can not be reliably estimated. Choosing too large windows, on the other hand, might mean that there is drift within windows, which in turn might result in an underestimation of the overall discrimination the classifier produces over the course of the stream as distinct P_{t_1}, P_{t_2} with $P_{t_1} \neq P_{t_2}$ are sampled from. Additionally, overlaps between windows can unintentionally give more weight

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

to individual samples, which are evaluated twice and therefore unintentionally receive a higher weight when computing the score for the overall stream.

All these problems combined create an effect where window size and overlaps have a substantial influence on the final fairness score computed over them, which leads to noticeable variance in fairness scores (e.g., demographic parity) even for relatively small changes in window size and overlap. Thus, relying on the simple window-based approach does not yield a good drift-aware fairness evaluation.

4.2 Driftpoint-based approach

The challenge, therefore, consists of avoiding the pitfalls of purely cumulative scores over the entire stream, while also not falling prey to the issue of arbitrary window sizes and overlaps. Assuming an ideal world, we would know the drift characteristics, being able to split the data stream into non-drifting segments, which then could be analyzed individually and accumulated afterwards. This maximizes windowlengths, allowing us to estimate each P_t as well as possible, while also avoiding the issue of mixing different $P_{t_1} \neq P_{t_2}$ in the same window. However, in realistic settings, we do not really know when fairness-relevant drift occurs. Thus, we propose to rely on an unsupervised change-point detector analyzing the probability distribution of protected attributes and labels to approximate the ideal setting. Here, the only hyperparameter choice is the number of expected changepoints or their significance level for automatic estimation, as well as the features for which drift is analyzed. This minimizes the risk of fairness-hacking through hyperparameter selection we have with window sizes and overlaps.

4.3 Experiments

Using the *Kernel-based Changepoint Detector* by [7], we evaluate this approach on modified versions of the *Adult* and *Student* datasets based on [6], where drift was introduced by mixing different streams. As [6] only introduces datasets with one driftpoint, we modify those by introducing a few more splits—between two and four—in order to evaluate our drift detection models.

For our experiments, we employed the kernel-based drift detector only on an extremely reduced number of features, namely labels and sensitive attributes, in order to minimize the influence of noise. We also evaluated both the effects of detecting a number of different changepoints, as well as using an algorithm [8] to determine the number of relevant drifts occurring in a stream. We run our experiments ten times, with the original data from which the drifting streams are created randomly permuted, comparing the score based on the drift-detection windows with both the "ideal" score based on the ground truth information of the drift, as well as the score computed over the full stream (the current state of the art). Our drift point detector uses a Gaussian kernel, and the automatic estimation of the number of driftpoints uses a slope heuristic hyperparameter of $\alpha = 3$, as this resulted in stable predictions for our streams, and a more thorough investigation into drift-detection is out of scope for this paper [1].

[1] Code available on GitHub

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

Table 1: Fairness scores accumulated over the full stream (SOTA), our detected drift windows, and ideal, ground truth drift windows .

Fairness Score		**Datasets**			
		Adult		Student	
		Gender Switch	**Debiasing**	**Boy Support**	**Adjust Grades**
Demographic Parity	Full Stream	3.12 ±0.40	0.19 ±0.14	1.04 ±0.52	0.74 ±0.53
	Drift Windows	28.42 ±1.19	3.49 ±0.51	1.04 ±0.52	2.40 ±0.74
	Ideal Windows	28.43 ±1.20	3.53 ±0.53	2.07 ±0.60	2.40 ±0.76
Disparate Impact	Full Stream	6.71 ±3.72	0.37 ±0.37	1.14 ±0.58	0.86 ±0.63
	Drift Windows	101.99 ±74.04	7.10 ±3.77	1.14 ±0.58	2.82 ±0.97
	Ideal Windows	102.13 ±74.16	7.18 ±3.82	2.26 ±0.69	2.81 ±0.99
Equal Opportunity	Full Stream	4.98 ±3.96	1.47 ±1.67	1.03 ±0.59	0.91 ±0.64
	Drift Windows	74.97 ±71.53	6.55 ±3.20	1.03 ±0.59	3.08 ±1.02
	Ideal Windows	75.07 ±71.64	6.57 ±3.26	2.06 ±0.70	3.08 ±1.06
Equal FPR	Full Stream	4.45 ±3.55	2.14 ±1.85	2.73 ±3.18	1.32 ±1.18
	Drift Windows	60.97 ±66.52	5.84 ±3.03	2.73 ±3.18	3.86 ±1.72
	Ideal Windows	61.05 ±66.63	5.86 ±3.08	3.67 ±3.02	3.85 ± 1.73

As can be seen in Table 1, our experiments show that the kernel-based drift detection method is relatively reliable in finding drift that affects the model's fairness in a highly significant manner. However, on the "Boy Support" stream, which had the least strong drift, no potential driftpoints met the relevancy-threshold we set for the automatic estimation of the number of driftpoints. In this case, the drift-detection score and the current SOTA, the fully cumulative score, are therefore identical. In the other scenarios, where drift was more pronounced and detectable, the scores computed by our drift-detection method were very close to the ground truth, and the scores differed very little between ideal splits and splits made by the detector.

This shows the relative robustness of the drift-detection approach with automatic estimation of the number of changepoints—drifts with strong effects on the fairness score are reliably detected well, ensuring that drastic drift effects are not overlooked, while smaller effects are potentially ignored in favour of the cumulative score, eliminating the issue of the effects of arbitrary window sizes, which might be more pronounced on smaller or more imbalanced streams.

Additionally, we also experimented with the effect of the number of changepoints themselves as a hyperparameter for the drift detection method [7] without automatic estimation, also computing how well the actual changepoints were represented (see Fig. 2), visualizing the furthest minimal distance between real and detected points. Our results show that on the long, strongly drifting stream *Adult Gender Switch*, changepoints are swiftly detected, and additional changepoints have little influence on the resulting score. On the smaller *Student Grade Scale Change* stream with lower baseline fairness scores and less fairness-relevant drift, true changepoints are reliably detected, but additional splits have a relatively strong influence on score computation. This shows that over-estimating the number of changepoints can be unproblematic in some scenarios, with high baseline unfairness and longer drift-free windows as well as stronger drift, but also very problematic in other settings. Using the algorithm in [8] might therefore be better, although further research into this issue would be prudent to

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964.
Available from http://www.i6doc.com/en/.

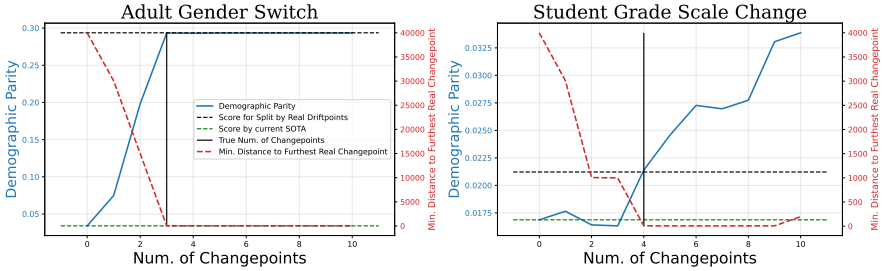

Fig. 2: The effect of the number of changepoints on the demographic parity score.

develop clear, practical guidelines.

5 Conclusion

In this paper, we discussed the need for a suitable evaluation protocol for fairness in drifting data streams, explored two potential evaluation schemes, and demonstrated the suitability of the drift-point-based evaluation. But while this is an important first step to evaluating fairness in drifting datastreams, more work is required. For instance, it would be useful to analyze how well the proposed evaluation strategy can capture more subtle changes occurring over a longer time, how these measures could be used for monitoring or adaptive fairness constraints. Explicitly time-sensitive fairness notions might also be of interest.

References

[1] D. Pessach and E. Shmueli. A review on fairness in machine learning. *ACM Comput. Surv.*, feb 2022.

[2] V. Iosifidis and E. Ntoutsi. Fabboo - online fairness-aware learning under class imbalance. In *IFIP Working Conference on Database Semantics*, 2020.

[3] V. Iosifidis et. al. Fairness-enhancing interventions in stream classification. In *Database and Expert Systems Applications*. Springer, 2019.

[4] F. Hinder et. al. One or two things we know about concept drift, part a. *Frontiers in Artificial Intelligence*, 2024.

[5] Jacob Montiel et. al. River: machine learning for streaming data in python. *J. Mach. Learn. Res.*, 2021.

[6] K. Lammers et. al. Realistic benchmarks for fair stream learning. *Neural Information Processing*, 2025.

[7] Z. Harchaoui and O. Cappé. Retrospective mutiple change-point estimation with kernels. *14th Workshop on Statistical Signal Processing*, 2007.

[8] S. Arlot et. al. A kernel multiple change-point algorithm via model selection. *Journal of Machine Learning Research*, 2019.