SHORT-PAPER

Interactive Fairness Auditing: Leveraging AVOIR for Dynamic Evaluation and Mitigation

AMIN MEGHRAZI, The Ohio State University, Columbus, OH, United States

PRANAV MANERIKER, The Ohio State University, Columbus, OH, United States

SWATI PADHEE, The Ohio State University, Columbus, OH, United States

SRINIVASAN PARTHASARATHY, The Ohio State University, Columbus, OH, United States

Open Access Support provided by:

The Ohio State University

Published: 22 June 2025

Citation in BibTeX format

SIGMOD/PODS '25: International Conference on Management of Data
June 22 - 27, 2025
Berlin, Germany

Conference Sponsors:
SIGMOD

https://doi.org/10.1145/3722212.3725108
ISBN: 9798400715648

Interactive Fairness Auditing: Leveraging AVOIR for Dynamic Evaluation and Mitigation

Amin Meghrazi
meghrazi.1@osu.edu
The Ohio State University
Columbus, OH, USA

Pranav Maneriker
maneriker.1@osu.edu
The Ohio State University
Columbus, OH, USA

Swati Padhee
padhee.3@osu.edu
The Ohio State University
Columbus, OH, USA

Srinivasan Parthasarathy
srini@cse.ohio-state.edu
The Ohio State University
Columbus, OH, USA

Abstract

Ensuring fairness in high-stakes machine learning (ML) applications remains a significant challenge. We introduce an interactive, Streamlit-based User Interface (UI) for AVOIR, an automated fairness monitoring framework. This UI allows users to select fairness metrics (e.g., Demographic Parity, Equalized Odds) and explore dataset attributes through dynamic visualizations. The UI supports iterative refinement, enabling users to define fairness constraints in a domain-specific language (DSL), evaluate models, and address violations via interactive charts. Leveraging AVOIR's inference and optimization capabilities, it provides probabilistic guarantees and detects biases at runtime. We demonstrate its utility on multiple datasets, showing how fairness violations are identified and mitigated. By combining declarative fairness specifications, rigorous optimization, and intuitive visualizations, our UI empowers stakeholders to deploy ML models responsibly.

CCS Concepts

• **Human-centered computing → Visualization design and evaluation methods**.

Keywords

Fairness Auditing, Interactive User Interface, Responsible AI

ACM Reference Format:
Amin Meghrazi, Pranav Maneriker, Swati Padhee, and Srinivasan Parthasarathy. 2025. Interactive Fairness Auditing: Leveraging AVOIR for Dynamic Evaluation and Mitigation. In *Companion of the 2025 International Conference on Management of Data (SIGMOD-Companion '25), June 22–27, 2025, Berlin, Germany*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3722212.3725108

1 Introduction

The rapid proliferation of machine learning (ML) technologies has brought them into high-stakes domains such as healthcare, finance, and criminal justice [1]. While these systems offer significant benefits, they also raise critical concerns about fairness and accountability. Many models operate as opaque black boxes, making their decision-making processes challenging to interpret and leaving them vulnerable to biases that disproportionately affect marginalized groups. Consequently, there is an increasing demand for tools to ensure fairness in ML models, particularly in legally and ethically sensitive contexts.

Auditing fairness in real-time for complex, black-box models poses significant challenges. Traditional methods are resource-intensive, requiring large datasets and manual intervention, making them unsuitable for continuous monitoring [2]. Recent efforts have focused on theoretical frameworks investigating both the design of metrics and their probabilistic quantification to help audit the biases of models [3–6]. From a practical standpoint, end-users and stakeholders evaluating algorithms for biases may not have the expertise to leverage these frameworks, as they require experience with probabilistic specifications and concentration guarantees. Therefore, it is essential to provide them with simple visual interfaces that enable them to leverage the insights provided by such frameworks. Leveraging AVOIR [6], which enables auditing in both ML and database settings, we introduce an interactive, user-friendly interface for real-time fairness evaluation and refinement. Developed with Streamlit[1], our User Interface (UI) enables users to visualize fairness metrics across diverse datasets dynamically. Users can define and adjust fairness constraints to assess and monitor models, including black-box systems, ensuring adaptability and transparency.

A key feature of our UI is its support for iterative refinement of fairness metrics. Users can define fairness criteria, detect violations through interactive charts, and adjust specifications on the fly. Leveraging AVOIR's probabilistic guarantees, the UI offers real-time insights into fairness violations and enables dynamic adjustments to investigate mechanisms to alleviate such violations. We also demonstrate the seamless integration of AVOIR within database materialized views. We demonstrate the utility of AVOIR UI using the Adult Income [7], COMPAS [8] datasets and can apply it to folkables/ACS-Income [9] to show its broader applicability. This work highlights the practical benefits of integrating declarative fairness specifications with probabilistic inference and intuitive visualizations, enabling stakeholders to deploy ML models responsibly while ensuring fairness and transparency.

[1] https://streamlit.io/

SIGMOD-Companion '25, Berlin, Germany

ACM ISBN 979-8-4007-1564-8/2025/06
https://doi.org/10.1145/3722212.3725108

2 Definitions and Preliminaries

This section defines the core concepts of AVOIR, starting with Fairness Specifications, followed by an overview of fairness notions in the UI and the AVOIR framework.

2.1 Fairness Specification

In AVOIR, a **Fairness Specification** is a formal expression that defines the fairness criterion an ML model must satisfy. It specifies how the model's output (represented by the variable r) is expected to behave across different groups defined by sensitive attributes, such as race or gender. Users can define these specifications using a domain-specific language (DSL), which provides a structured and flexible framework for expressing fairness constraints. A fairness specification in AVOIR consists of three components:
(i) **Expectation Terms**: The fundamental components of a fairness specification are the *Expectation Terms* (E), which represent the expected values of the model's output variable r under different conditions. For example, $E(r \mid G = g_1)$ represents the expected outcome for a group $G = g_1$, where G could be a sensitive attribute such as gender or race. (ii) **Threshold**: A threshold is a user-defined value that specifies the maximum allowable deviation between the expected outcomes of different groups, determining whether a fairness criterion is met. (iii) **Output Variable (r)**: The output variable r represents the predicted outcome of the model. In a fairness specification, r is the variable whose expected value is compared across groups.

2.2 Fairness Measures

We now briefly review how three popular fairness measures used in the ML literature can be defined within the AVOIR framework [6]:.
Demographic Parity: This fairness criterion ensures that the proportion of individuals receiving a positive outcome is equal across two groups. It is specified as:

$$\frac{E(r \mid G = g_1)}{E(r \mid G = g_2)} < \text{threshold}$$

Equalized Odds: This fairness criterion requires equal error rates for both labels (false positives and false negatives) across different groups. It is expressed as:

$$\frac{E(r \mid G = g_1 \wedge Y = y)}{E(r \mid G = g_2 \wedge Y = y)} < \text{threshold}$$

where Y represents the true outcome (e.g., whether someone re-offended in the case of recidivism risk prediction) for both $y \in \{0, 1\}$.
Equal Opportunity: A weaker version of Equalized Odds, this criterion focuses on equalizing the true positive rate across different groups. It is formulated as:

$$\frac{E(r \mid G = g_1 \wedge Y = 1)}{E(r \mid G = g_2 \wedge Y = 1)} > \text{threshold}$$

2.3 AVOIR Framework Overview

AVOIR is a framework for real-time monitoring of ML model fairness, providing continuous updates and probabilistic guarantees of compliance with predefined specifications. The key components and steps involved in AVOIR's operation are as follows:
Inference Engine: The inference engine calculates the expected values of the outcome variable r for different groups. Using model output data, it updates these expectations in real-time to ensure ongoing monitoring of fairness specifications. Additionally, the engine propagates uncertainty across fairness components, adjusting as more data is collected.
Failure Probability: AVOIR assigns a failure probability (δ) to each fairness specification, representing the likelihood of a fairness criterion being violated based on the observed values of inputs and outputs. The framework aims to minimize this probability adaptively, given the data observed.
Adaptive Optimization: AVOIR uses an *adaptive optimization* technique to adjust the allocation of failure probabilities across different terms within a specification to minimize the overall failure probability. This novel strategy allows the system to focus resources on the most uncertain aspects, ensuring fairness violations are detected early, even with limited data.

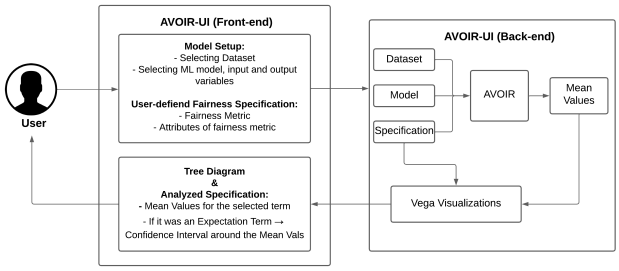

Figure 1: AVOIR-User Interface Diagram

3 AVOIR-UI Architecture

Figure 1 illustrates the pipeline for the AVOIR UI, designed to allow users to define, evaluate, and refine fairness specifications for ML models. The architecture comprises several key components that interact with one another, enabling a seamless user experience for fairness monitoring and evaluation.

3.1 Model Setup

As the first step in using the AVOIR UI, users need to set up the model, as illustrated in Figure 2, as follows:
Selecting Dataset: Users can select a dataset from a dropdown for model evaluation and fairness analysis.
Selecting ML Model: Next, users can select an ML model to predict the output variable. The interface includes six models: linear regression, decision tree, logistic regression, RBF-SVM classifier, three-layer MLP, and BERT-based text regression. Users must define the input variables (features) and the output variable (target), as accurate specification is crucial for reliable fairness analysis.

3.2 User-Defined Fairness Specification

Once the model setup is complete, users can define their fairness specifications. This component allows users to interact with the system by selecting the fairness metric to evaluate and specify the parameters that define the fairness condition. The main functionalities are as follows:
Selecting Fairness Metric: The system supports multiple fairness metrics, such as *Demographic Parity*, *Equalized Odds*, and *Equal*

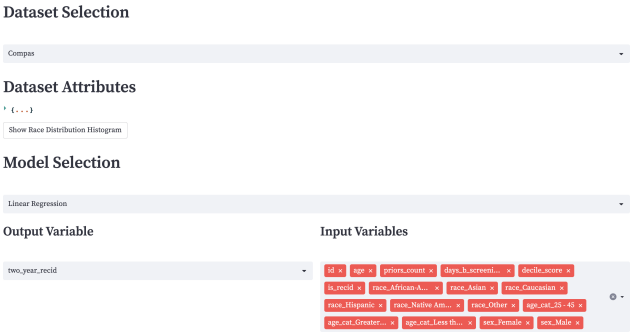

Figure 2: Model Setup Interface

Figure 3: User-Defined Fairness Specification Interface

Opportunity. Users can choose one of these metrics based on the fairness property they wish to monitor.

Defining Attributes for the Fairness Metric: After selecting a fairness metric, users must specify its attributes. For instance, with *Demographic Parity*, users define the groups they want to compare (e.g., race, gender) and set a threshold value to determine the acceptable difference in expected outcomes between these groups. For *Equalized Odds* or *Equal Opportunity*, the relevant groups and conditions are similarly specified.

These fairness constraints are represented using a domain-specific language (DSL), ensuring they are valid and syntactically correct before proceeding with the analysis. Figure 3 illustrates where fairness metrics are selected, groups are customized, and the generated specifications are displayed.

3.3 Vega Integration

After loading the model and evaluating the user-defined fairness specifications, we use Vega[2], a declarative JSON-based grammar, to map data to visual elements for generating visualizations. During AVOIR runs, the system logs the evaluation values and corresponding observation indices, which are converted into a structured tabular JSON format (each row represents a specific grammar element and its corresponding metrics). The Vega specification processes the structured data to create interactive, web-based plots, ensuring a clean separation between evaluation and visualization while maintaining integration with the back-end workflow.

[2]https://vega.github.io/vega/

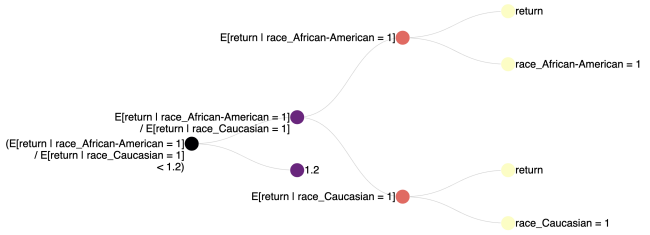

Figure 4: Tree Diagram Corresponded to the Generated Fairness Specification

3.4 Visualizations

Building on the Vega-based approach discussed above, the following components handle the data transformation and rendering processes, ensuring that fairness insights are conveyed clearly and intuitively to end-users.

Tree Diagram: The tree diagram corresponds to the syntax tree of the parsed DSL and provides a visual representation of the fairness specification. It shows how the different components of the fairness criterion are related, helping users understand the structure of the specification and offering transparency on how fairness is assessed. As shown in Figure 4, the tree diagram illustrates the relationships and conditions contributing to the fairness assessment.

Analyzed Specification: After evaluation, the system presents the results of the fairness analysis, including the mean values for the selected node in the tree diagram, and the confidence interval around the mean values if the selected node corresponds to an expectation term. This helps identify fairness issues and supports iterative refinements.

As shown in Figure 5, the bottom chart shows the confidence interval for an expectation term, while the top chart displays only the mean values, as it represents the division of two expectation terms. AVOIR incurs comparable runtimes across fairness metrics that is generally lower for Adult Income dataset than COMPAS, which is expected given COMPAS is about six times larger. Among the metrics, Equalized Odds being the most complex typically takes the longest to compute.

3.5 Materialized View Support

We expect to demonstrate the integration of the AVOIR-UI with database materialized views.We use *pandas dataframes* [10] to simulate the application of AVOIR in the database setting. Specifically, we wrap pandas dataframes with a Python 'Database' class and provide a query mechanism that outputs materialized views. Queries are implemented through Python functions that take a dataframe as input and output a result set dataframe.The corresponding view thus generated monitors the base pandas dataframe for insertion of new data through a refresh function. The monitored specification is added as a decorator inside the refresh function, allowing AVOIR to track specifications for the materialized view.Note that we discuss the integration with *pandas* only for ease of exposition; AVOIR's

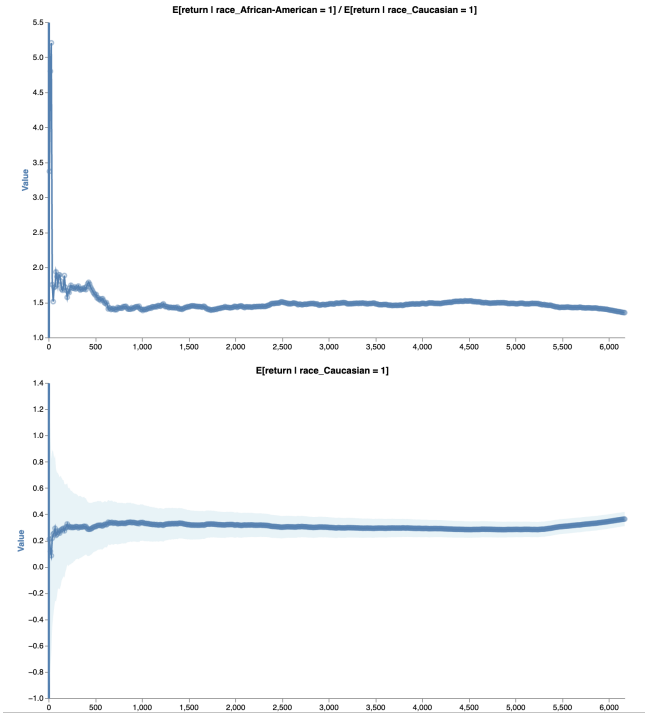

Figure 5: Value Charts for Selected Tree Nodes: The x-axis shows timestamped data points as observed by AVOIR, and the y-axis shows the cumulative mean up to each point.

inference engine and optimization can be integrated into any open-source or commercial database engine natively or via user-defined functions (UDFs).

4 Audience Experience

While dataset selection, model setup, and visualization were previously discussed, this demonstration emphasizes how end-users can actively audit the fairness of models through our UI. We present a case study using the COMPAS dataset, as shown in Figure 5. In this example, the plots correspond to the Demographic Parity concept

$$\frac{\mathbb{E}[\text{return} \mid \text{race\_African-American} = 1]}{\mathbb{E}[\text{return} \mid \text{race\_Caucasian} = 1]},$$

while also displaying confidence intervals for *each subgroup's* expected outcomes (rather than for the ratio itself). These intervals help users assess the stability or uncertainty of each group's average outcome. A wide interval suggests significant variability or limited data, encouraging users to gather more evidence before drawing firm conclusions. In contrast, a narrow interval indicates more consistent estimates for that subgroup's outcome. To interpret these metrics, attendees can set a threshold (e.g., 1.2). If the observed ratio remains below this threshold, the model is considered fair under the selected criterion; otherwise, it may indicate a potential fairness violation.

Attendees can adjust fairness thresholds or switch fairness notions (e.g., Equal Opportunity), triggering real-time updates to visualizations. This allows them to identify disparities and assess whether the model meets the selected fairness criterion, demonstrating how our UI enables interactive fairness auditing.

5 Potential Impact

The development of AVOIR and its integration into a user-friendly interface can significantly enhance the fairness and accountability of ML models in high-stakes decision-making systems. By enabling real-time monitoring and iterative refinement of fairness specifications, this interface empowers stakeholders to identify and mitigate biases that may disproportionately affect marginalized groups. This approach fosters greater transparency in ML, ensuring that models deployed in critical areas like healthcare, finance, and criminal justice adhere to ethical standards and regulatory requirements. The ability to dynamically adjust fairness constraints based on evolving data also enables continuous improvement of model fairness, promoting more equitable and responsible use of ML technologies. As part of ongoing work, we hope to extend AVOIR to accommodate graph-based representation learning tasks[11].

Acknowledgments

The authors acknowledge support from the National Science Foundation (NSF) under grant 2112471 (AI-EDGE), a research grant from Cisco (US202581249) and support from Ohio Cyber Range Institute under grant (Grant-AWD-117076). They are also deeply grateful to Aditya Vadlamani for his valuable feedback on this work. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] Tiago P Pagano, Rafael B Loureiro, Fernanda VN Lisboa, Rodrigo M Peixoto, Guilherme AS Guimarães, Gustavo OR Cruz, Maira M Araujo, Lucas L Santos, Marco AS Cruz, Ewerton LS Oliveira, et al. Bias and unfairness in machine learning models: a systematic review on datasets, tools, fairness metrics, and identification and mitigation methods. *Big data and cognitive computing*, 7(1):15, 2023.

[2] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé, Miro Dudik, and Hanna Wallach. Improving fairness in machine learning systems: What do industry practitioners need? In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–16, New York, NY, USA, 2019. Association for Computing Machinery.

[3] Bishwamittra Ghosh, Debabrota Basu, and Kuldeep S Meel. Justicia: A stochastic sat approach to formally verify fairness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7554–7563, 2021.

[4] Bing Sun, Jun Sun, Ting Dai, and Lijun Zhang. Probabilistic verification of neural networks against group fairness. In *Formal Methods: 24th International Symposium, FM 2021, Virtual Event, November 20–26, 2021, Proceedings 24*, pages 83–102. Springer, 2021.

[5] Tom Yan and Chicheng Zhang. Active fairness auditing. In *International Conference on Machine Learning*, pages 24929–24962. PMLR, 2022.

[6] Pranav Maneriker, Codi Burley, and Srinivasan Parthasarathy. Online fairness auditing through iterative refinement. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 1665–1676, New York, NY, USA, 2023. Association for Computing Machinery.

[7] D. Dua and C. Graff. Adult income dataset. https://archive.ics.uci.edu/ml/datasets/adult, 2019. UCI Machine Learning Repository.

[8] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing, 2016. ProPublica.

[9] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. *Advances in neural information processing systems*, 34:6478–6490, 2021.

[10] The pandas development team. pandas-dev/pandas: Pandas, February 2020.

[11] A. Vadlamani, A. Srinivasan, P. Maneriker, A. Payani, and S. Parthasarathy. A generic framework for conformal fairness. In *To appear in ICLR*, 2025.