FairX: A comprehensive benchmarking tool for model analysis using fairness, utility, and explainability

Md Fahim Sikder[1,*], Resmi Ramachandranpillai[2], Daniel de Leng[1] and Fredrik Heintz[1]

[1]*Department of Computer and Information Science (IDA), Linköping University, Sweden*
[2]*Institute for Experiential AI, Northeastern University, USA*

Abstract

We present FairX, an open-source Python-based benchmarking tool designed for the comprehensive analysis of models under the umbrella of fairness, utility, and eXplainability (XAI). FairX enables users to train benchmarking bias-mitigation models and evaluate their fairness using a wide array of fairness metrics, data utility metrics, and generate explanations for model predictions, all within a unified framework. Existing benchmarking tools do not have the way to evaluate synthetic data generated from fair generative models, also they do not have the support for training fair generative models either. In FairX, we add fair generative models in the collection of our fair-model library (pre-processing, in-processing, post-processing) and evaluation metrics for evaluating the quality of synthetic fair data. This version of FairX supports both tabular and image datasets. It also allows users to provide their own custom datasets. The open-source FairX benchmarking package is publicly available at https://github.com/fahim-sikder/FairX.

Keywords

Fair evaluation, Benchmarking tool, Synthetic data, Data utility, Explainability

1. Introduction

With the rapid development of artificial intelligence-based systems to aid us in our daily lives, it is important for these systems to give outcomes that is acceptable for all users, including—but not limited to—from demographic perspective. Troublingly, as the available data is filled with human or machine bias, models trained with these dataset often gives unfair outcome towards some demographic [1]. It is therefore critical to mitigate bias in the dataset and model. Over the years, researchers have used different techniques to achieve this [2, 3]. These techniques can be roughly grouped into three families: 1) *Pre-processing*, i.e. where the dataset is processed in such a manner that it produces less biased outcomes, before passing it to a model for training; 2) *In-processing*, i.e. where the model learns the original data distribution and shifts the data distribution to a fair distribution by adding some constraints during the training process; and 3) *Post-processing*, i.e. where the model's outcome is changed in such a manner that it gives fair outcomes relative to protected attributes. The performance of these models or datasets can be

AEQUITAS 2024: Workshop on Fairness and Bias in AI | co-located with ECAI 2024, Santiago de Compostela, Spain
*Corresponding author.
md.fahim.sikder@liu.se (M. F. Sikder); r.ramachandranpillai@northeastern.edu (R. Ramachandranpillai); daniel.de.leng@liu.se (D. de Leng); fredrik.heintz@liu.se (F. Heintz)
https://fahimsikder.com/ (M. F. Sikder)
0000-0001-5307-997X (M. F. Sikder)

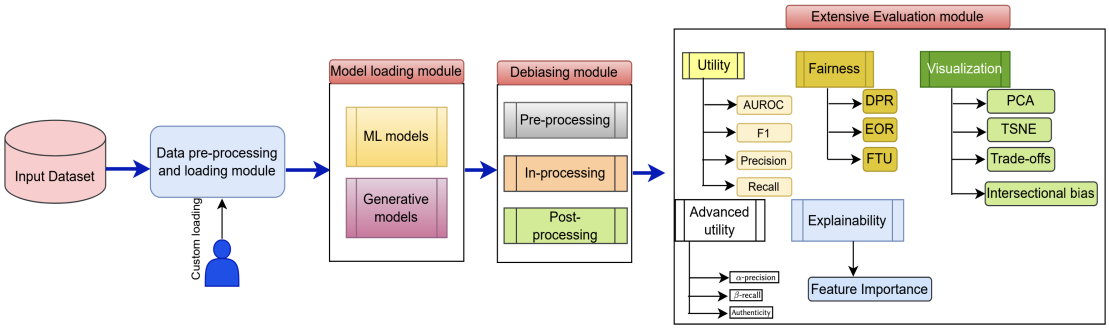

Figure 1: A High-level overview of FairX. An input dataset (possibly custom) is fed to the FairX data loading module followed by a bias-mitigation module and an extensive evaluation module providing multi-faceted evaluations.

measured by the evaluation metrics that reflect both the fairness and data utility. To ease up the work for training models and evaluating them, researchers has developed benchmarking tool that bring the training and evaluation in one framework. Recently, research on fair generative models has found a lot of spotlight and measuring the quality of the synthetic data is as crucial as evaluating fairness and data utlity.

Existing fairness-related benchmarking tools focus on creating benchmarks and measuring their fairness on different datasets. For example, FairLearn [4] by Microsoft contains several fair models and evaluation metrics for checking fairness and data utility. AI Fairness 360 (AIF360) [5] by IBM also contains fairness evaluation metrics and basic data utility measuring metrics. But both of these frameworks lack the ability to train fair generative models and measure the data utility for synthetic data. For synthetic fair data, it is important to validate the quality of the generated data alongside measuring the fairness and other data utilities. Explainability is an essential property of fair models because it aids in making the model's decision-making process more transparent. These modules should therefore be included in such benchmarking tools.

In this work, we present *FairX*, an open-source modular fairness benchmarking tool, available to use at https://github.com/fahim-sikder/FairX. A high-level system overview is given in Figure 1. FairX contains data processing techniques and benchmarking fairness models (incorporating pre-processing, in-processing, and post-processing), including generative fair models. We evaluate these models in terms of fairness, data utility. We also add evaluation methods for synthetic fair data (Advanced Utility) to check the quality of the generated samples. FairX supports both tabular and image data and can plot feature importance for down-streaming task using explainable algorithms.

The remainder of this paper is organised as follows. In Section 2 we discuss some background information that will help the reader understand the rest of the paper. We then present FairX in Section 3. Section 4 shows some fairness results obtained by FairX for a number of datasets and models. Finally, the paper looks ahead towards future improvements in Section 5.

2. Background

In this section, we provide the necessary details to follow the paper.

2.1. Bias mitigation methods

A variety of bias mitigation methods have been proposed in the literature based on data, training, and predictions. These methods can be broadly categorized into three main approaches: pre-processing, in-processing, and post-processing techniques.

Pre-processing. These techniques involve altering the training data to resolve any potential causes of biases before it is fed to the model. There are various techniques in the literature such as disparate impact remover [6], data cleaning and augmentation, and fair representation learning [7]. This involves balancing the representation of different groups or generating synthetic data to augment underrepresented groups, assigning weights to uphold some minority groups, and transforming the data representation in a format that obscures protected features while maintaining feature attributions.

In-processing. This involves mitigating biases during training. The techniques involve fairness constraints, adversarial de-biasing [8], and fairness-aware learning. In fairness constraints training, a multi-objective optimization combining a prediction loss and a fairness penalty will be used such as adding regularization terms to the objective function that penalizes unfairness or incorporating fairness metrics as part of the optimization process. In adversarial de-biasing [8], adversarial training is used to reduce bias. The model is trained to perform well on the primary classification/prediction tasks while simultaneously trying to prevent an adversary from predicting the protected features, thus forcing the model to learn less biased representations.

Post-processing. These methods are applied to the predictions of a classifier. Techniques such as threshold adjustment, calibration [9], and Reject Option Classifications [10] fall under this category. In threshold adjustment the decision thresholds of a trained model are adjusted to ensure that the outcomes meet the chosen fairness metric. Calibration [9] ensures that the predicted probabilities maintain the true likelihood of outcomes equally across different demographic groups. Techniques like equalized odds post-processing is used where the model's outputs are adjusted to satisfy fairness constraints. Reject Option-Based Classification (ROC) [10] allows the model to prevent from making a decision when the confidence is low, for the chosen sensitive attributes. This can reduce the likelihood of biased or unfair decisions in uncertain instances.

2.2. Evaluation metrics

To measure the performance of models or dataset, various evaluation methods are being used. For evaluating fair model or checking the dataset for potential bias, different kinds of fairness metrics exists. For example, demographic parity checks if the decision from a down-streaming task is equal for each class in sensitive attributes. Fairness through unawareness [11] checks

Table 1
Comparison of existing benchmarking tools with FairX over different key areas of interests: Fairness Evaluation; Synthetic Data Evaluation; Model Explainability; and Generative Fair Model Training.

Benchmarking Tools	Fairness Evaluation	Synthetic Data Evaluation	Explainability	Generative Model Training
Fairlearn [4]	✓	✗	✗	✗
AIF360 [5]	✓	✗	✓	✗
Jurity [14]	✓	✗	✗	✗
AEQUITAS [15]	✓	✗	✗	✗
REVISE [17]	✓	✗	✗	✗
FairBench [16]	✓	✗	✗	✗
FairX (ours)	✓	✓	✓	✓

how the accuracy of down-stream task effects if no-sensitive attributes is used during the training and prediction phase. Adding fairness constraints to the models or datasets may change the data distributions and thereby affect the performance of the dataset or models [12]. To check the data utility performance, we commonly use Accuracy score, F1-score, Precision and Recall. To evaluate the quality of the synthetic data researchers use, α-precision [13], β-recall [13]. Also to check, is the generative model is truly generating new contents or not, the metrics authenticity [13] is being used.

2.3. Comparison of existing benchmarking tools

Over the years researchers have developed various fairness benchmarking tools which commonly include a dataset loader, different bias mitigation techniques and evaluation metrics. Fairlearn [4] by Microsoft is one such benchmarking tool. It has support for different algorithms for bias mitigation and measuring the fairness of a model. AIF360 [5] by IBM is another benchmarking tool. It supports a wide range of evaluation metrics (both for fairness and data utility) and bias-removal algorithms (in-processing, pre-processing and post-processing). Another example is Jurity [14]. It contains recommender system evaluations, and various fairness and data utility functions. AEQUITAS [15], FairBench [16] generate fairnes report and REVISE [17] is a tool to detect and mitigate bias in the image dataset. More recently, in the area of generative models, there has been an increased interest in generating fair data in the image, tabular and medical domains [18, 19, 20, 1, 21, 22]. But the aforementioned benchmarking tools do not contain these models. Also, when evaluating models, other benchmarking tools, only measure the fairness and data utility of the models itself. But evaluation methods for generated data is needed. We need to verify the quality of the synthetic data. We also need to verify the authenticity of the synthetic data, to show the generative models are actually generating new content rather than just copying the data itself. FairX is bridging this gap. We add support for evaluating synthetic data and add generative models in our benchmarking tool. Table 1 shows the comparison among the models with FairX.

3. FairX

In this section we present FairX in detail. FairX is built on three primary modules, 1) the *Data Loading Module*, 2) the *Bias-mitigating Techniques Module*, and 3) the *Evaluation Module*. The main pipeline (shown in Figure 1) works as follows. Given a dataset, FairX will pre-process it in a way that is compatible with the benchmarking model. Next the model will train itself using the dataset. After the training the evaluation module will give the results based on fairness, data utility and explain the outcome using explainability.

3.1. Data loading module

The BaseDataClass handles the internal processing of datasets and make it compatible with the bias-mitigating models that are present on our framework as well as making it easier to handle for other bias-mitigating models that are not present in this tool. This class contains different methods for handling different kinds of data extension (CSV, and others). We add three widely used tabular datasets (Adult-Income, COMPAS and Credit Card) and two image datasets (Colored MNIST and CelebA) in the benchmarking tool, and we plan to add more. The BaseDataClass process datasets based on numerical and categorical features. It also provides methods to normalize the dataset and is equipped with functionality for various encodings (e.g. One-hot encoding, QuantileTransformer). It also has a dataset-splitting function to split the dataset for training and testing purpose. We also add functionality to prepare the dataset for explainability algorithms. Sample usage of datasets are described in Appendix Section A, Listing 1.

Custom Dataset Loader. Besides adding widely used benchmarking datasets for fair data research, we also provide the option to use custom dataset. By using the CustomDataClass, users can load their own dataset (CSV, TXT, etc.) and train the models. Users need to specify the sensitive attributes and target attributes while using the CustomDataClass. Pre-processing and other functionalities are also available in this class, like in the BaseDataClass. We present sample usage of CustomDataClass in Listing 4 of Appendix Section A.

3.2. Bias-mitigating techniques module

One of FairX's main aims is to benchmark different bias-mitigation techniques on various datasets. Over the years, different techniques have been proposed, and we add models from these techniques to the tool. For the benchmarking process, we use the same hyper-parameters used in their respective works. We create a common format for all the bias-mitigation techniques to make it easy for the users. For example, each bias-mitigation technique has its own class, which has model.fit() function. This fit() function takes the dataset and processes it (if needed for the specific model). For the generative models (in-processing techniques), this function also generates synthetic data and saves it as a Pandas dataframe. Sample usage of models is described in Appendix Section A, Listing 2.

Table 2
Breakdown of FairX-supported features.

Dataset		Adult-Income, Compas, Credit-card Colored MNIST (Image) CelebA (Image)
Models	Pre-processing	Correlation Remover
	In-processing	TabFairGAN [21] FairDisco [1] Decaf [22]
	Post-processing	Threshold Optimizer
Metrics	Fairness	Demographic Parity Ratio (DPR) Equilized Odds Ratio (EOR) Fairness through Unawareness (FTU)
	Data Utility	AUROC, F1-score, Precision Recall, Accuracy
	Synthetic Data Evaluation	α-precision [13] β-recall [13] Authenticity [13]
Plotting		PCA [23] & t-SNE [24] plots Feature Importance Fairness vs Accuracy Intersectional Bias
Explainability		Explain prediction of a model Feature Importance

Pre-processing. We add the support for Correlation remover [4] (CorrRemover in FairX) in the benchmarking. Correlation Remover removes the correlation between the sensitive attributes with other data features by using a linear transformation while keeping as much information as possible. It is also possible to control on how much correlation we want remove by using the remove_intensity parameter while the value 1.0 will result maximum correlation removal while 0.0 will do the opposite. We can access the pre-processing algorithm by using fairx.models.preprocessing.

In-processing. Most recent in-processing bias mitigation techniques are based on generative models. And the fairness benchmarking tools we mentioned in this work does not contain these models. One of our contribution of FairX is that, we add several fair generative models, such as, TabFairGAN [21], Decaf [22] and Fairdisco [1]. We can access the in-processing algorithm by using fairx.models.inprocessing module. After training, these models will generate and save the samples automatically.

Post-processing. For the post-processing bias mitigation technique, We add Threshold Optimizer [4]. This technique operates on a classifier and improve the output of its based on a fairness constraint. In this case, we use demographic_parity as a fairness constraint to improve the outcome of the classifier as presented in [4]. For using the post-processing algorithm, we can use fairx.models.postprocessing module.

3.3. Evaluation module

In FairX, we aim to evaluate the performance of model or dataset using wide range of evaluation metrics. We evaluate in terms of fairness, data utility. Other existing fairness benchmarking tools, lacks the capability to measure the data quality of the synthetic data. It is necessary to check the data quality of the synthetic data as well as the fairness criteria. Here, we present the evaluation module FairX has and we use *XGBoost* as a classifier, also we keep the option to use scikit-learn's LogisticRegression.

Fairness Evaluation. We create the FairnessUtils class to accommodate fairness evaluation metrics. In this class, currently we add the support for checking the Demographic Parity Ratio, Equalized Odds Ratio, Fairness Through Unawareness (FTU) metrics. We also have plan to add more metrics over the time. Fairness metrics can be accessed using the fairx.metrics.FairnessUtils module.

Data Utility. Beside checking the fairness criteria of the datasets or models, we also add the functionality to check the data utility using FairX. We add the support for checking the Accuracy, Precision, Recall, AUROC, and F1-score. And these functions can be accessed by using the fairx.metrics.DataUtilsMetrics module.

Synthetic Data Evaluation. In FairX, we add the functionality to evaluate the quality of the generated data by the fair generative models. It is important to validate the quality of the synthetic data along with the validation of fairness and data utility criteria. Existing fairness measuring benchmark do not have the functionality to evaluate the synthetic data quality. We evaluate the synthetic data quality in terms of fidelity, diversity and check if the synthetic data has any trace of original data in it [25]. We use α-precision [13] to evaluate the fidelity of the synthetic data, β-recall [13] to check the diversity and Authenticity [13] is used to check if the generative models are just memorising the training data or not. Synthetic data evaluation module can be accessed from fairx.metrics.SyntheticEvaluation. We also add the t-SNE and PCA plots to check the fidelity and diversity of the synthetic data too, more about the plots are discussed in section 3.4.

Explainability. We add the explainability functionality in FairX to explain the prediction of a model. We train a classifier (XGBoost) on the benchmarking datasets, and then we explain the prediction using the fairx.explainability.ExplainUtils module. This module is based on the TreeExplainer of SHAP [26]. Beside this, we give the functionality to show the feature importance while making a decision. This functionality is especially useful when we want to see how much importance is given to the sensitive attributes while making a decision.

3.4. Plotting

We add various plotting support in FairX. They can be accessed under the fairx.utils.plotting module. We add support to show the performance trade-off of model accuracy and their fairness performance. Also, we plot the feature importance to show which features are responsible for

Table 3
Evaluation on the **Adult-Income** dataset using different models presented at the FairX. Bold indicates best result, and all the metrics score are higher as better. Synthetic Data Evaluation is only applicable to the Fair Generative Models (i.e. TabFairGAN and Decaf).

		Fairness Metrics		**Data Utility**			**Synthetic Data Evaluation**		
	Protected Attribute	DPR	EOR	ACC	AUC	F1-Score	α-precision	β-recall	Authenticity
Correlation-Remover	Gender	0.32 ± .01	0.23 ± .01	0.86 ± .01	0.79 ± .01	0.71 ± .01	n/a	n/a	n/a
	Race	0.29 ± .01	0.20 ± .01	0.86 ± .01	**0.80 ± .01**	0.71 ± .01	n/a	n/a	n/a
TabFairGAN	Gender	0.69 ± .01	0.60 ± .01	0.84± .01	0.76 ± .01	0.65 ± .01	**0.91 ± .01**	**0.50 ± .01**	**0.537 ± .01**
	Race	0.026 ± .01	0.00 ± .00	0.84 ± .01	0.77 ± .01	0.67 ± .01	**0.98 ± .01**	**0.489 ± .01**	**0.52 ± .01**
Decaf	Gender	0.52 ± .01	0.42 ± .01	0.75 ± .01	0.63 ± .01	0.44 ± .01	0.07 ± .01	0.08 ± .01	0.009 ± .01
	Race	0.55 ± .01	0.46 ± .01	0.77 ± .01	0.69 ± .01	0.53 ± .01	0.06 ± .01	0.08 ± .01	0.009 ± .01
FairDisco	Gender	**0.98 ± .01**	**0.85 ± .01**	0.78 ± .01	0.63 ± .01	**0.86 ± .01**	n/a	n/a	n/a
	Race	**0.95 ± .01**	**0.92 ± .01**	0.812 ± .01	0.71 ± .01	**0.88 ± .01**	n/a	n/a	n/a
Threshold Optimizer	Gender	0.95 ± .01	0.35 ± .01	0.86 ± .01	0.66 ± .01	0.65 ± .01	n/a	n/a	n/a
	Race	0.69 ± .01	0.25 ± .01	0.87 ± .01	0.66 ± .01	0.71 ± .01	n/a	n/a	n/a
Original Data	Gender	0.32 ± .01	0.22 ± .01	**0.88 ± .01**	**0.80 ± .01**	0.72 ± .01	n/a	n/a	n/a
	Race	0.19 ± .01	0.00 ± .00	**0.88 ± .01**	**0.80 ± .01**	0.71 ± .01	n/a	n/a	n/a

Table 4
Evaluation on the **Compas** dataset using different models presented at the FairX. Bold indicates best result, and all the metrics score are higher as better. Synthetic Data Evaluation is only applicable to the Fair Generative Models (i.e. TabFairGAN and Decaf).

		Fairness Metrics		**Data Utility**			**Synthetic Data Evaluation**		
	Protected Attribute	DPR	EOR	ACC	AUC	F1-Score	α-precision	β-recall	Authenticity
Correlation-Remover	Gender	0.43 ± .01	0.33 ± .01	0.64 ± .01	0.64 ± .01	0.59 ± .01	n/a	n/a	n/a
	Race	0.58 ± .01	0.63 ± .01	0.65 ± .01	0.64 ± .01	0.60 ± .01	n/a	n/a	n/a
TabFairGAN	Gender	0.52 ± .01	0.42 ± .01	**0.68 ± .01**	**0.68 ± .01**	**0.66 ± .01**	**0.84 ± .01**	**0.70 ± .01**	0.37 ± .01
	Race	0.50 ± .01	0.49 ± .01	**0.69 ± .01**	**0.68 ± .01**	**0.64 ± .01**	**0.94 ± .01**	**0.75 ± .01**	0.33 ± .01
Decaf	Gender	0.87 ± .01	0.84 ± .01	0.45 ± .01	0.45 ± .01	0.42 ± .01	0.77 ± .01	0.45 ± .01	**0.61 ± .01**
	Race	**0.99 ± .01**	**0.96 ± .01**	0.45 ± .01	0.45 ± .01	0.42 ± .01	0.77 ± .01	0.45 ± .01	**0.61 ± .01**
FairDisco	Gender	**0.97 ± .01**	0.92 ± .01	0.55 ± .01	0.54 ± .01	0.43 ± .01	n/a	n/a	n/a
	Race	0.87 ± .01	0.76 ± .01	0.53 ± .01	0.53 ± .01	0.44 ± .01	n/a	n/a	n/a
Threshold Optimizer	Gender	0.92 ± .01	**0.98 ± .01**	0.65 ± .01	0.65 ± .01	0.61 ± .01	n/a	n/a	n/a
	Race	**0.99 ± .01**	0.76 ± .01	0.63 ± .01	0.63 ± .01	0.60 ± .01	n/a	n/a	n/a
Original Data	Gender	0.37 ± .01	0.28 ± .01	0.66 ± .01	0.65 ± .01	0.61 ± .01	n/a	n/a	n/a
	Race	0.54 ± .01	0.58 ± .01	0.66 ± .01	0.65 ± .01	0.61 ± .01	n/a	n/a	n/a

prediction outcome. This comes in handy analyzing original data and synthetic fair data to see how much the fair model reduce the feature importance for the sensitive attributes.

To show the quality of the synthetic data generated by the fair generative models, we add PCA and t-SNE plots. These plots shows how close the synthetic data is from the original data.

4. Results and discussion

We now consider the fairness, data utility and synthetic data evaluation (only for in-processing generative models) of the models presented in this benchmarking tool. We also present the

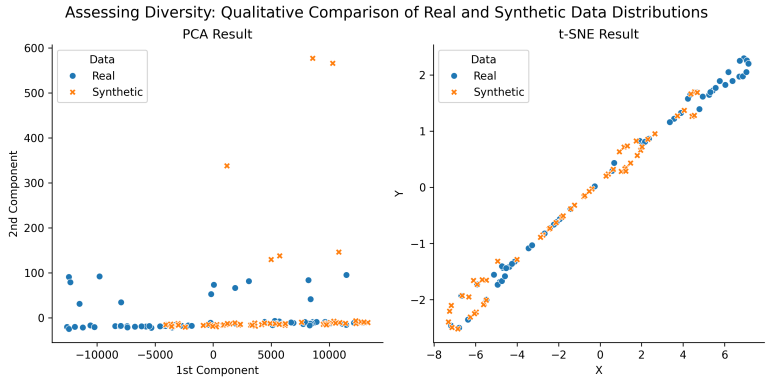

Figure 2: PCA and t-SNE plots of the original data and Synthetic data generated by TabFairGAN. Here each dot represents a record, if the generative model learns the original data distribution then the dots should overlap with each other. Dataset: 'Adult-Income', Protective attribute: 'sex'.

explainability analysis where we use the generated data by in-processing generative models and show how the fair generated data perform on down-streaming task and how the prediction is affected by the sensitive attributes. We also show the feature importance by using these explainability analysis.

Table 3 and 4 shows the performance of the bias mitigation algorithms for the *Adult-Income* dataset and *Compas* dataset respectively. We run experiment using different Protected attributes [1]. Besides, fairness and data utility, we add synthetic data evaluation for the output of TabFairGAN [2], and Decaf [3].

From the table, we see for the generative fair models, TabFairGAN is performing well comparing with the Decaf in both datasets with both protected attributes. The $\alpha-$precision, $\beta-$recall scores of TabFairGAN is better than Decaf, this represents the synthetic data quality of TabFairGAN is superior than Decaf. On the other hand, TabFairGAN perform poorly in the fairness evaluation for the 'race' protected attribute of the *Adult-Income* dataset. Whereas In-processing technique FairDisco [4] performs well in terms of fairness and data utility.

On the visual evaluation of fair synthetic data, we use the synthetic data generated by TabFairGAN. Figure 2 shows the PCA and t-SNE plots of the synthetic data generated by TabFairGAN. We show how closely the synthetic data distribution is matching with the original data. If the generative model can capture the original data distribution, original and synthetic data should overlap with each other on the PCA and t-SNE plot. Figure 2 shows that data generated by TabFairGAN partially learned the data distribution of the original data.

In Figure 3, we show the feature importance for a down-streaming task to predict the target attribute of the *Adult-Income* dataset where the 'Sensitive attribute' is 'sex'. We compared the

[1]For the sake of brevity, we could not include additional results using other datasets—we refer the reader to the FairX repository for these results. Some metrics like precision, recall, fairness through unawareness (FTU), and plots like fairness-accuracy trade-offs were similarly omitted.

[2]https://github.com/amirarsalan90/TabFairGAN

[3]https://github.com/vanderschaarlab/synthcity

[4]https://github.com/SoftWiser-group/FairDisCo

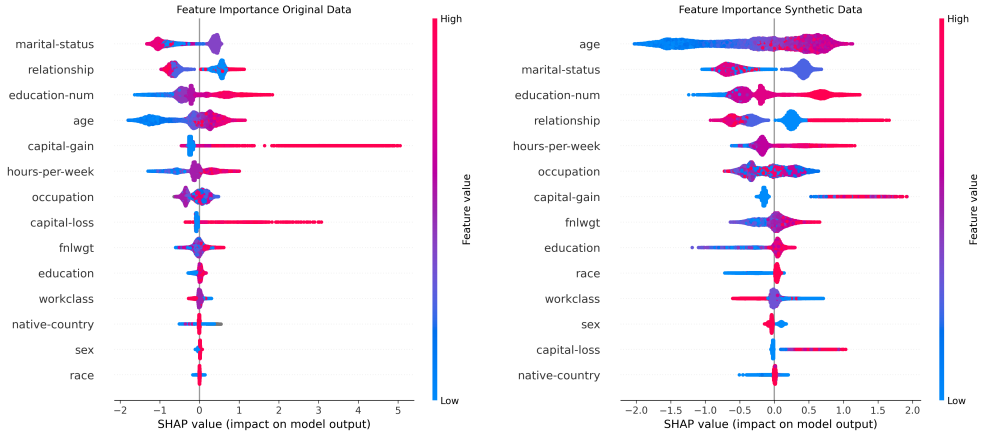

Figure 3: Feature Importance on Prediction task on the Original Data (left) and Synthetic Data (right) by TabFairGAN, the Sensitive Feature here is 'sex', The Feature Value of Sensitive Attribute in Synthetic Data is less than Original Data.

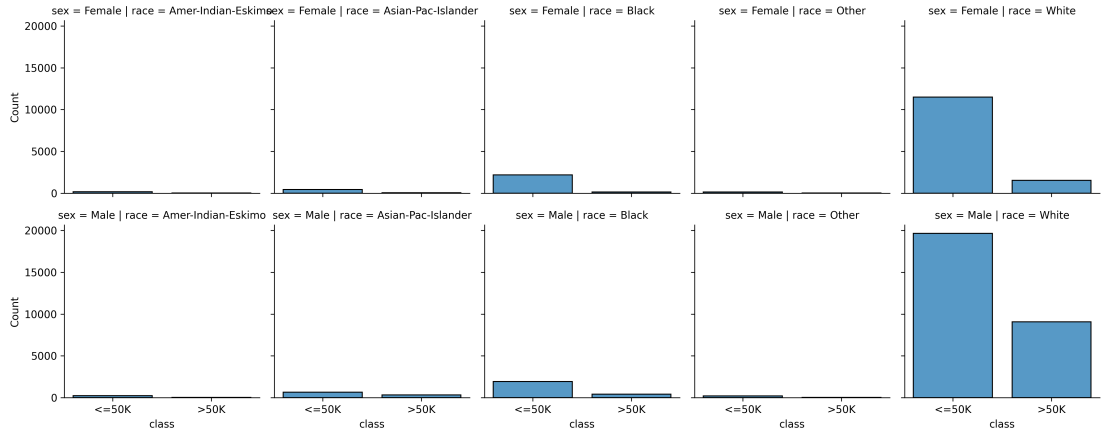

Figure 4: Representation of 'sex' and 'race' features on the target class, here we can see the dataset is heavily in favor of white people.

feature importance of original data with the synthetic data generated by TabFairGAN. We can see the feature importance of the synthetic data is lower than the original data. This means the synthetic data generated by the TabFairGAN is less biased towards entity.

Finally, Figure 4 shows the intersectional bias on the *Adult-Income* dataset. We plot the percentage of 'salary-income' for both 'race' and 'sex' protected attributes. We see in the dataset, decisions are given in favor towards white people.

5. Conclusion and future work

Massive amounts of data are being produced everyday. Unfortunately, much of this data contains human or machine biases. Furthermore, the usage of recommendation system has increased with advancements in artificial intelligence. But if we use biased data to train a recommendation system, there is a high chance that the recommendation system will yield unfair decision towards some demographics. To mitigate this issue, researchers have developed various measure to mitigate the bias from the dataset, or to train the model in such a way that the model produces bias-free data. To help in this process, benchmarking tools equipped with different bias-mitigation techniques and evaluation metrics were developed over the years. But these benchmarking tools commonly lack the option to evaluate generative models or to train them. We therefore presented FairX, an open-source, modular, fairness benchmarking tool. FairX comes with a data-loader, supports model training, and has an evaluation module. FairX provides support for training fair generative models and for evaluating the synthetic data created by them. FairX also contains various fairness evaluation metrics, data utility evaluation metrics and different plotting techniques to help users to evaluate models and visualize outcomes. FairX comes with support for explainability analysis of a prediction using the dataset (both original and synthetic) and shows feature importance. We believe FairX will help the researchers and mitigate the gap of not having fair generative models and way of evaluating synthetic data.

In the future, we intend to extend FairX to be able to handle other modalities in addition to tabular and image data, for example text and video. Also, we will add wider range of evaluation metrics for both synthetic data utility and fairness metrics. For the models, we plan to add text based and more tabular and image based fair generative models [19, 20, 27, 18]. In this version of FairX, we do not have option to add custom models, but we plan to add this features in future version, so users can use their own model and use all the functionalities of FairX for their model. We also plan to add hyper-parameter optimization feature for the models so, we can find the optimal parameters and best result. Finally, we plan to add functionalities to evaluate the output of large language models.

Acknowledgments

The work was partially funded by the Knut and Alice Wallenberg Foundation, and the TAILOR Network of Excellence for trustworthy AI (EC Grant Agreement 952215). Portions of this work were carried out using the AIOps/Stellar facilities funded by the Excellence Center at Linköping–Lund in Information Technology (ELLIIT).

References

[1] J. Liu, Z. Li, Y. Yao, F. Xu, X. Ma, M. Xu, H. Tong, Fair representation learning: An alternative to mutual information, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 1088–1097.

[2] E. Ntoutsi, P. Fafalios, U. Gadiraju, V. Iosifidis, W. Nejdl, M.-E. Vidal, S. Ruggieri, F. Turini, S. Papadopoulos, E. Krasanakis, et al., Bias in data-driven artificial intelligence systems—an

introductory survey, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 10 (2020) e1356.
[3] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan, A survey on bias and fairness in machine learning, ACM computing surveys (CSUR) 54 (2021) 1–35.
[4] H. Weerts, M. Dudík, R. Edgar, A. Jalali, R. Lutz, M. Madaio, Fairlearn: Assessing and improving fairness of ai systems, Journal of Machine Learning Research 24 (2023) 1–8.
[5] R. K. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilović, et al., Ai fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias, IBM Journal of Research and Development 63 (2019) 4–1.
[6] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, S. Venkatasubramanian, Certifying and removing disparate impact, in: proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, 2015, pp. 259–268.
[7] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, C. Dwork, Learning fair representations, in: International conference on machine learning, PMLR, 2013, pp. 325–333.
[8] B. H. Zhang, B. Lemoine, M. Mitchell, Mitigating unwanted biases with adversarial learning, in: Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, 2018, pp. 335–340.
[9] G. Pleiss, M. Raghavan, F. Wu, J. Kleinberg, K. Q. Weinberger, On fairness and calibration, Advances in neural information processing systems 30 (2017).
[10] F. Kamiran, A. Karim, X. Zhang, Decision theory for discrimination-aware classification, in: 2012 IEEE 12th international conference on data mining, IEEE, 2012, pp. 924–929.
[11] G. Cornacchia, V. W. Anelli, G. M. Biancofiore, F. Narducci, C. Pomo, A. Ragone, E. Di Sciascio, Auditing fairness under unawareness through counterfactual reasoning, Information Processing & Management 60 (2023) 103224.
[12] J. Wang, X. E. Wang, Y. Liu, Understanding instance-level impact of fairness constraints, in: International Conference on Machine Learning, PMLR, 2022, pp. 23114–23130.
[13] A. Alaa, B. Van Breugel, E. S. Saveliev, M. van der Schaar, How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models, in: International Conference on Machine Learning, PMLR, 2022, pp. 290–306.
[14] M. Thielbar, S. Kadıoğlu, C. Zhang, R. Pack, L. Dannull, Surrogate membership for inferred metrics in fairness evaluation, in: International Conference on Learning and Intelligent Optimization, Springer, 2023, pp. 424–442.
[15] P. Saleiro, B. Kuester, L. Hinkson, J. London, A. Stevens, A. Anisfeld, K. T. Rodolfa, R. Ghani, Aequitas: A bias and fairness audit toolkit, arXiv preprint arXiv:1811.05577 (2018).
[16] E. Krasanakis, S. Papadopoulos, Towards standardizing ai bias exploration, arXiv preprint arXiv:2405.19022 (2024).
[17] A. Wang, A. Narayanan, O. Russakovsky, REVISE: A tool for measuring and mitigating bias in visual datasets, in: European Conference on Computer Vision (ECCV), 2020.
[18] R. Ramachandranpillai, M. F. Sikder, D. Bergström, F. Heintz, Bt-GAN: Generating Fair Synthetic Healthdata via Bias-transforming Generative Adversarial Networks, Journal of Artificial Intelligence Research (JAIR) 79 (2024) 1313–1341.
[19] J. Li, Y. Ren, K. Deng, Fairgan: Gans-based fairness-aware learning for recommendations with implicit feedback, in: Proceedings of the ACM web conference 2022, 2022, pp. 297–307.
[20] R. Ramachandranpillai, M. F. Sikder, F. Heintz, Fair Latent Deep Generative Models

(FLDGMs) for Syntax-Agnostic and Fair Synthetic Data Generation, in: ECAI 2023, IOS Press, 2023, pp. 1938–1945.
[21] A. Rajabi, O. O. Garibay, Tabfairgan: Fair tabular data generation with generative adversarial networks, Machine Learning and Knowledge Extraction 4 (2022) 488–501.
[22] B. Van Breugel, T. Kyono, J. Berrevoets, M. Van der Schaar, Decaf: Generating fair synthetic data using causally-aware generative networks, Advances in Neural Information Processing Systems 34 (2021) 22221–22233.
[23] F. B. Bryant, P. R. Yarnold, Principal-Components Analysis and Exploratory and Confirmatory Factor Analysis (1995).
[24] L. Van der Maaten, G. Hinton, Visualizing Data using T-SNE, Journal of machine learning research 9 (2008).
[25] M. F. Sikder, R. Ramachandranpillai, F. Heintz, Transfusion: Generating long, high fidelity time series using diffusion models with transformers, arXiv preprint arXiv:2307.12667 (2023).
[26] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, S.-I. Lee, From local explanations to global understanding with explainable ai for trees, Nature Machine Intelligence 2 (2020) 2522–5839.
[27] K. Choi, A. Grover, T. Singh, R. Shu, S. Ermon, Fair generative modeling via weak supervision, in: International Conference on Machine Learning, PMLR, 2020, pp. 1887–1898.

A. Detailed Usage

In this section, we present different sample code example of our tool. We give a brief description of each module and their corresponding class description and function details.

Dataset usage. To use the dataset already pre-loaded with the tool, we need to use the BaseDataClass. This class takes three hyperparameters as input; dataset_name, sensitive_attirbute and a boolean flag for attaching the target variable with the main dataframe. BaseDataClass has two functions, preprocess_data() and split_data() to preprocess the dataset using categorical, numerical transformation and split the dataset for training and testing purpose respectively.

```
from fairX.dataset import BaseDataClass

dataset_name = 'Adult-Income'
sensitive_attribute = 'race'
attach_target = True
data_module = BaseDataClass(dataset_name, sensitive_attribute, attach_target)

```

Listing 1: Using BaseDataset Class.

Model usage. We add three kinds of bias-removal techniques under the models folder of FairX. The list of available models can be found in Table 2. Here is an example usage of in-processing algorithm called *TabFairGAN*. After initializing the Model, we train the it by calling

the FIT() function which takes the dataset, batch size and number of epochs as parameters. After training, for the fair generative models (TabFairGAN and Decaf), synthetic data will be automatically saved in the working directory.

```
    from fairX.models.inprocessing import TabFairGAN

    data_module = BaseDataClass(dataset_name, sensitive_attribute, attach_target)
    under_prev = 'Female'
    y_desire = '>50K'
    tabfairgan = TabFairGAN(under_prev, y_desire)
    tabfairgan.fit(data_module, batch_size = 256, epochs = 1000)

```

Listing 2: Using Models.

Metrics usage. Here, we give a sample code for measuring the fairness and data utilities with a dataset that is already part of the FairX system. Both FAIRNESSUTILS and DATAUTILSMETRICS class takes the dataset as input and then we call the EVALUATE_FAIRNESS() and EVALUATE_UTILITY() function to measure the fairness data utilities respectively. The result is stored as a dictionary file.

```
    from fairX.metrics import FairnessUtils
    from fairX.metrics import DataUtilsMetrics
    from fairX.dataset import BaseDataClass

    data_module = BaseDataClass(dataset_name, sensitive_attribute, attach_target)
    cat_transformer, num_scaler, transformed_data = data_module.preprocess_data()
    splitted_data = data_module.split_data(transformed_data)
    fairness_measurement = FairnessUtils(splitted_data)
    utility_measurement = DataUtilsMetrics(splitted_data)
    fairness_res = fairness_measurement.evaluate_fairness()
    datautils_res = utility_measurement.evaluate_utility()
    print(fairness_res)
    print(datautils_res)

```

Listing 3: Using Fairness & Data utility Metrics.

The following code example is to use the CUSTOMDATACLASS to load custom dataset in FairX. We need to give the dataset path, list of sensitive attributes and a boolean operator for attaching the target. This code also shows the usage of synthetic data evaluation using the SYNTHETICEVALUATION class.

```
    from fairX.metrics import SyntheticEvaluation
    from fairX.dataset import BaseDataClass
    from fairX.dataset import CustomDataClass

    original_data = BaseDataClass(dataset_name, sensitive_attribute, attach_target)
    generated_data = CustomDataClass(generated_data_path, sensitive_attribute,
attach_target)
    synthetic_evaluation_class = SyntheticEvaluation(original_data, generated_data)
    synthetic_data_measurement = synthetic_evaluation_class.evaluate_synthetic()
    print(synthetic_data_measurement)
```

Listing 4: Using Synthetic Data Evaluation Metrics with Custom Data Loader.