MPDS

Final Report

Exploiting Bias in Data Through Conditional Approximate Functional Dependencies

Mining and Profiling Data Streams

Gianmario Voria
gvoria@unisa.it

July 17, 2024

Computer Science Department
University of Salerno

Contents

1 Introduction 2

2 Background and Related Work 3

3 Preliminary Notions 3
3.1 Functional Dependencies 3
3.2 Conditional Functional Dependencies 3
3.3 Approximate Functional Dependencies 4

4 Data Preparation and Conditions Application 4
4.1 Step 1. Dataset Selection 4
4.2 Step 2. Data Preparation 5
4.3 Conditions Application 5

5 AFD Discovery 6
5.1 Results 6

6 Fairness Metrics Computation 7

7 Analysis of the Results and Concluding Remarks 8

References 9

1 Introduction

In the era of big data and artificial intelligence, the integrity and fairness of data-driven decision-making processes have come under increased scrutiny. As datasets grow in size and complexity, the potential for bias and discrimination embedded within them becomes a significant concern. This research project aims to address these issues by analyzing Approximate Conditional Functional Dependencies (ACFDs) in datasets to detect possible bias and discrimination and subsequently computing fairness metrics to verify the discovered bias issues within the realm of software engineering.

Conditional Functional Dependencies (CFDs) are powerful tools used to express constraints that must hold under specific conditions in a dataset. They extend traditional Functional Dependencies (FDs) by incorporating conditions, making them particularly suitable for capturing complex real-world scenarios [1, 2]. However, in many practical applications, data may be incomplete, noisy, or exhibit inherent uncertainties, making it necessary to relax these constraints. Approximate Conditional Functional Dependencies (ACFDs) offer a solution by allowing for dependencies that hold for a majority, but not necessarily all, of the data instances [3, 4]. This flexibility is essential for identifying patterns and anomalies that indicate potential bias and discrimination.

Bias in datasets can manifest in various forms, such as disparate impact, where certain groups are disadvantaged by decision-making algorithms, or disparate treatment, where different groups are treated unequally based on discriminatory attributes. Detecting such biases requires sophisticated techniques that can navigate the nuances of real-world data. By leveraging ACFDs, this research aims to uncover hidden biases that might not be apparent through traditional analysis methods.

Once potential biases are detected, it is crucial to quantify their impact. Fairness metrics provide a means to evaluate the degree of bias in a dataset or algorithm [5, 6]. These metrics are essential for assessing the fairness of software systems and ensuring they comply with ethical standards and regulatory requirements. In the context of software engineering, fairness metrics help verify that the algorithms and systems developed are not perpetuating or amplifying existing biases [7].

The primary objectives of this research are:

- To analyze ACFDs in unfair datasets to identify potential bias and discrimination.
- To compute and evaluate fairness metrics that quantify the extent of detected biases.

2 Background and Related Work

Machine Learning Fairness. The problem of ensuring machine learning fairness has been tackled from different perspectives. Researchers have been investigating data diversity as the underlying driver of fairness, starting from the hypothesis that **discrimination arises from the ML applications being trained on biased or unbalanced datasets** [8]. On the one hand, Zhang and Harman [9] claimed that having a dataset with many features does not help reduce discrimination; on the other hand, Chakraborty et al. [10] showed that the selection of relevant features and data heavily influences the biased outcomes. These observations pointed out that the data selection process is not trivial and needs specific attention to be properly executed. To this aim, Chakraborty et al. [10] designed *Fair-SMOTE*, a fair data balancing algorithm that does not negatively impact learning performance. Similarly, Moumoulidou et al. [11] augmented the *Max-Min* diversification objective with fairness constraints, proposing three innovative algorithms guaranteeing robust theoretical approximation tailored to varying combinations of parameters. Gilbert [12] explored algorithmic bias issues, fostering the creation of effective fair ML toolkits, while Caton and Haas [13] organized existing approaches and techniques to deal with fairness into *pre-processing*, *in-processing*, and *post-processing*, based on the phase they should be applied in.

Related Work. From a data profiling perspective, Azzalini, Criscuolo, and Tanca proposed E-FAIR-DB [14]. The system is designed to detect and mitigate bias in datasets using Approximate Conditional Functional Dependencies (ACFDs). The process begins with an investigation phase, during which the system applies a Data Bias discovery procedure. This phase uncovers dependencies that reveal any discrimination or privilege among groups within the dataset and assesses the dataset's level of diversity.

Following this, the user must decide if the discovered bias is relevant to their analysis. The system allows the user to train a model on the original dataset and evaluate it based on the intended analysis and the fairness issues identified during the Data Bias discovery step. This evaluation helps the user determine whether the identified bias impacts their analysis significantly.

The user can proceed to the ACFD-Repair procedure if the bias is deemed relevant. This step aims to mitigate the initial bias using the previously discovered ACFDs. The ACFD-Repair process can be repeated until the user is satisfied with the mitigation results. Once the user is content with the outcome, they can save the cleaned dataset.

Thus, The E-FAIR-DB system offers a comprehensive framework for detecting, assessing, and mitigating dataset bias, ultimately enhancing diversity and fairness in data-driven analyses.

3 Preliminary Notions

Understanding the various dependencies between data attributes is crucial in database systems for ensuring data integrity, optimizing query performance, and facilitating database design. This section explores three key concepts: functional dependencies, conditional functional dependencies, and approximate functional dependencies, which form the foundation for many database operations and design principles.

3.1 Functional Dependencies

Functional dependencies (FDs) are a fundamental concept in relational database theory, introduced by Edgar F. Codd. An FD is a constraint between two sets of attributes in a relation from a database. Specifically, a functional dependency, denoted as $X \to Y$, indicates that for any two tuples in the relation, if they agree on the attributes in set X, they must also agree on the attributes in set Y. This concept is pivotal in the normalization process, which aims to reduce data redundancy and improve data integrity by organizing the attributes of a database into tables according to certain rules.

3.2 Conditional Functional Dependencies

Conditional functional dependencies (CFDs) extend the notion of traditional FDs by incorporating conditions that must hold for the dependency to be applicable. CFDs allow for more expressive constraints by specifying conditions

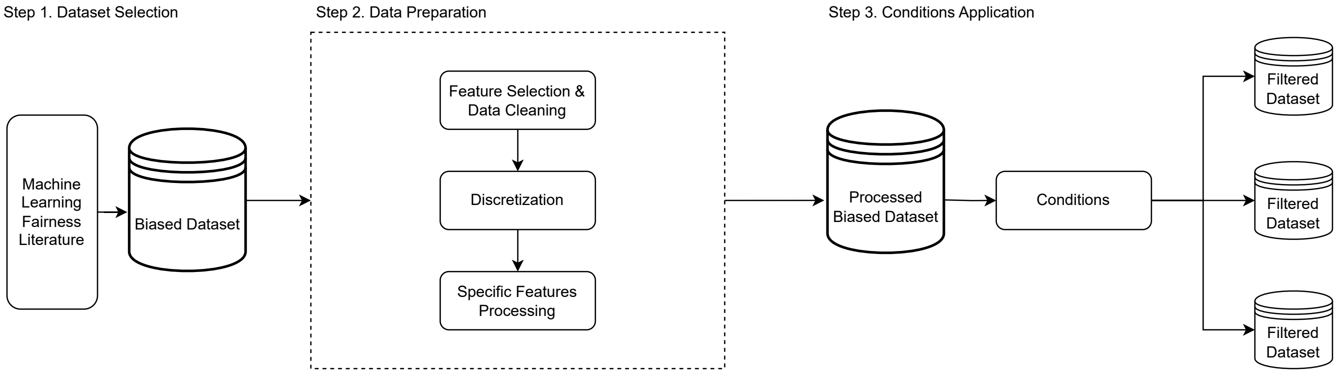
Step 1. Dataset Selection
Step 2. Data Preparation
Step 3. Conditions Application
Machine Learning Fairness Literature
Biased Dataset
Feature Selection & Data Cleaning
Discretization
Specific Features Processing
Processed Biased Dataset
Conditions
Filtered Dataset
Filtered Dataset
Filtered Dataset

Figure 1: Overview of the steps performed in the project's first phase.

under which the dependencies should be enforced. These are particularly useful in real-world scenarios where certain constraints are valid only under specific conditions. A CFD is typically expressed as $(X \to Y, \varphi)$, where φ is a condition expressed as a conjunction of predicates. Such conditional constraints are instrumental in ensuring data quality and are widely used in data cleaning and data integration tasks.

3.3 Approximate Functional Dependencies

Approximate functional dependencies (AFDs) relax the strictness of traditional FDs by allowing for dependencies that hold for a majority, but not necessarily all, of the tuples in a relation. This concept is useful in scenarios where data may be noisy or incomplete, often in large-scale databases and real-world applications. AFDs are particularly valuable in data mining and machine learning, where exact dependencies are rare, but approximate patterns can still provide significant insights. The use of AFDs can improve the efficiency of query processing and enhance the robustness of database systems by accommodating the inherent uncertainty and variability in real-world data.

4 Data Preparation and Conditions Application

Figure 1 summarizes the steps performed in this project's first phase. First, we selected an existing dataset widely employed in the fairness literature as the subject of the experiments. Afterward, we performed several Data Preparation practices following the design of E-FAIR-DB [14]. Finally, to simulate the conditions of CFDs, we applied 3 different conditions to the dataset's tuples and created 3 smaller datasets, each of which is only composed of tuples that respect the condition. The remainder of this section details each of the steps performed.

4.1 Step 1. Dataset Selection

Following the experiments performed in E-FAIR-DB [14], we selected the U.S. Census Adult Dataset [**adult**].

The U.S. Census Adult Dataset, also known as the "Adult" or "Census Income" dataset, is a widely-used dataset in machine learning and data analysis, particularly for classification and bias detection tasks. It was extracted from the 1994 Census database and is often used for benchmarking algorithms.

The dataset is designed to predict whether an individual's annual income exceeds 50,000 dollars based on various demographic and socioeconomic attributes. The dataset is available from the UCI Machine Learning Repository and contains the following features:

- Age: Continuous variable representing the age of the individual.
- Workclass: Categorical variable with several possible values indicating the individual's type of employment.
- Fnlwgt: Continuous variable representing the final weight, an estimation of the number of people the census entry represents.
- Education: Categorical variable indicating the highest level of education attained.

- Education-num: Continuous variable representing the number of years of education completed.
- Marital-status: Categorical variable indicating marital status.
- Occupation: Categorical variable with various possible values representing the individual's occupation.
- Relationship: Categorical variable representing the individual's relationship status.
- Race: Categorical variable.
- Sex: Categorical variable indicating gender: Male or Female.
- Capital-gain: Continuous variable representing capital gains.
- Capital-loss: Continuous variable representing capital losses.
- Hours-per-week: Continuous variable representing the weekly hours worked.
- Native-country: Categorical variable indicating the individual's native country.
- **Income**: Target feature, a binary class label indicating whether the individual's income exceeds 50,000 dollars per year.

The U.S. Census Adult Dataset is used to explore and model socioeconomic factors influencing income. It provides a rich set of features for analysis and algorithm development. Due to its diverse and comprehensive attribute set, it is particularly useful for studying income prediction, bias detection, and fairness in machine learning models.

4.2 Step 2. Data Preparation

In this step, we performed three different data preparation techniques, following the design of E-FAIR-DB [14]:

Feature Selection and Data Cleaning. The majority of the missing values belong to attributes that are not relevant to our analysis (e.g., "Marital-Status"), and therefore, we decided to perform feature selection first, then remove the few tuples that still contain missing values. The features removed were Marital-Status and Education-Num. Afterward, we proceeded with the removal of NA values and duplicate tuples.

Discretization To extract Functional Dependencies that are not reliant on specific values of a continuous or rational attribute, it is often beneficial to group the values into well-defined bins. Therefore, we performed discretization, a technique that converts continuous features or variables into discrete categories or bins. This method is commonly used in data preprocessing to transform continuous data into a format that can be more easily managed by various machine learning algorithms. In this step, we discretized two features: Hours-Per-Week and Age. Specifically, we established five intervals for the "Hours-Per-Week" attribute: "0–20," "21–40," "41–60," "61–80," and "81–100", and similarly, five intervals for "Age": "15–30," "31–45," "46–60," "61–75," and "76–100". Finally, we removed the original features and added the discretized ones to the dataset.

Specific Features Processing For clearer and more impactful Functional Dependencies, we retain the attribute "Race" in its original form from the dataset and categorize the values of the attribute "NC" into four distinct groups: "NC-US," "NC-Hispanic," "NC-Non-US-Hispanic," and "NC-Asian-Pacific". This was performed by manually mapping each unique country in the dataset to a specific group and then swapping the existing country feature with the relative group.

4.3 Conditions Application

In this step, we applied different conditions to the dataset's tuples. This creates a new dataset for each condition, which contains only the tuples for which the condition holds. The three conditions applied are:

- People that work in Private and have at least a Bachelor level of education, 6555 rows
- People that are from North America and are not married, 14739 rows
- People that are married, have no Bachelor/Masters/Doctoral degree and come from non American countries (europe, hispanic or asia), 1879 rows

5 AFD Discovery

For each dataset, the DiMe[15] algorithm was executed with two threshold configurations for the extent: 10% and 20% as the error tolerance limit in validation using the coverage measure g3-error.

The DiMe algorithm is used to discover approximate functional dependencies (AFDs). AFDs are useful in identifying relationships between attributes in datasets that may not be exact but hold true for most of the data, which is particularly important when dealing with noisy or incomplete datasets.

The extent in this context refers to the threshold for error tolerance. It defines the maximum allowable proportion of instances where the dependency can be violated while still considered valid. For instance, an extent of 10% means that up to 10% of the instances can violate the dependency, and it will still be recognized as an AFD.

The g3-error, or generalized error, is a coverage measure used to evaluate the validity of the discovered dependencies. It quantifies how well the dependency covers the dataset, considering the allowable extent. A lower g3-error indicates a higher quality of the discovered dependency, meaning it is more likely to be a true representation of the underlying relationship between the attributes.

By running the DiMe algorithm with these two extent thresholds, the analysis aims to identify the most robust and significant AFDs under different error tolerance levels, providing a comprehensive understanding of the dataset's attribute relationships.

5.1 Results

The computation of FDs provided insightful results in the possible presence of bias. The DiMe algorithm provided the results in Table 1.

Dataset	**Extent**	**Number of FDs**
Adult with Condition 1	10%	112
Adult with Condition 1	20%	108
Adult with Condition 2	10%	55
Adult with Condition 2	20%	152
Adult with Condition 3	10%	267
Adult with Condition 3	20%	481

Table 1: Functional Dependencies (FDs) discovered under different conditions and extent thresholds

With the computed AFDs, we proceeded with the analysis by filtering only dependencies that could lead to biased behavior. This was done by selecting the dependencies that matched the following condition:

Selection Condition: At least one protected attribute ('sex', 'race') on the left-hand side (LHS) and the target attribute ('income') on the right-hand side (RHS).

After this filtering, only two datasets with available dependencies were left, shown in Table 2. Only one dependency was left for the first dataset: ”fnlwgt, sex - income”. Table 3 shows the results for the second one.

Dataset	**Extent**	**Number of FDs**
Adult with Condition 2	10%	1
Adult with Condition 3	20%	20

Table 2: Datasets with available AFDs after the filtering process

AFD Discovery Results - We found out that in the Adult dataset that has been filtered only to contain tuples with people that are married, have no Bachelor/Masters/Doctoral degree, and come from non-American countries, several functional dependencies among protected attributes and the target attribute hold with an approximation extent of 20%. Hence, we will analyze this dataset to understand if the FDs discovered actually result in biased decisions by the machine learners.

LHS	RHS
workclass, education, occupation, relationship, race, capital-gain, age-range	income
education, occupation, race, age-range, hours-range, nc	income
workclass, education, occupation, relationship, race, capital-loss, hours-range, nc	income
education, occupation, sex, capital-loss, age-range, hours-range, nc	income
workclass, education, occupation, race, capital-gain, hours-range, nc	income
workclass, education, occupation, race, sex, capital-loss, hours-range, nc	income
education, occupation, race, capital-gain, capital-loss, age-range, nc	income
workclass, education, occupation, sex, capital-gain, capital-loss, hours-range, nc	income
workclass, education, occupation, sex, capital-loss, age-range, nc	income
workclass, education, occupation, sex, capital-gain, age-range, hours-range	income
education, occupation, race, capital-gain, capital-loss, age-range, hours-range	income
education, occupation, relationship, race, capital-gain, age-range, nc	income
workclass, education, occupation, race, capital-gain, capital-loss, age-range	income
education, occupation, race, sex, capital-gain, age-range, hours-range	income
education, occupation, race, sex, capital-gain, age-range, nc	income
workclass, education, occupation, race, age-range, hours-range	income
education, occupation, relationship, race, capital-gain, age-range, hours-range	income
education, occupation, sex, capital-gain, age-range, hours-range, nc	income
workclass, education, occupation, race, age-range, nc	income
workclass, education, occupation, race, sex, capital-gain, age-range	income

Table 3: AFDs in the dataset Adult with Condition 3, %20 extent.

6 Fairness Metrics Computation

We evaluated the fairness of machine learning models trained on the considered dataset to assess the identified bias and verify its severity.

First, we trained four algorithms such as *Logistic Regression* (LR), *Linear Support Vector Classification.* (SVC), *Random Forest* (RF), and *XGBoostClassifier* (XGB). These models were trained using all the features included in the datasets. As for the hyperparameters configuration, we used the default configurations.

Metric	Class	Precision	Recall	F1-score	Support	Precision	Recall	F1-score	Support
		Linear SVC				**XGBoost**			
Class	0	0.82	0.86	0.84	235	0.80	0.87	0.84	235
	1	0.74	0.70	0.72	141	0.75	0.64	0.69	141
Overall	Accuracy	0.80				0.78			
	Macro avg	0.78	0.78	0.78	376	0.78	0.76	0.76	376
	Weighted avg	0.79	0.80	0.79	376	0.78	0.78	0.78	376
		Random Forest				**Logistic Regression**			
Class	0	0.81	0.89	0.85	235	0.82	0.85	0.84	235
	1	0.78	0.66	0.71	141	0.74	0.70	0.72	141
Overall	Accuracy	0.80				0.79			
	Macro avg	0.79	0.77	0.78	376	0.78	0.77	0.78	376
	Weighted avg	0.80	0.80	0.80	376	0.79	0.79	0.79	376

Table 4: Combined Classification Reports

After training these models, we gather metrics and measures from the software engineering realm to evaluate their fairness level. These metrics are usually computed during the model's evaluation, and they verify the extent to which the predictions of such models are fair with respect to protected attributes and unprivileged groups.

To do this, we first identified the protected attributes (e.g., race, sex), privileged (e.g., white people, male), and unprivileged groups (e.g., black people, female).

Afterward, we calculated fairness metrics using the `IBM AIF-360` library:

- **Average Odds Difference (AOD)**: Average difference in False Positive Rate and True Positive Rate for unprivileged and privileged groups.
- **Statistical Partity Difference (SPD)**: This metric quantifies the disparity in predicted positive outcomes between a privileged group and an unprivileged group within a predictive model.
- **Equal Opportunity Difference (EOD)**: Measures the deviation from the equality of opportunity, which means that the same proportion of each population receives the favorable outcome.

Metric	XGBoost	Linear SVC	Random Forest	Logistic Regression
Sex SPD	-0.112	-0.175	-0.167	-0.178
Sex EOD	-0.267	-0.33	-0.29	-0.33
Sex AOD	-0.14	-0.203	-0.188	-0.206
Race SPD	0.18	0.145	0.18	0.141
Race EOD	0.016	0.014	0.042	0.014
Race AOD	0.109	0.075	0.109	0.072
Overall SPD	-0.064	-0.115	-0.118	-0.119
Overall EOD	-0.23	-0.293	-0.242	-0.293
Overall AOD	-0.108	-0.162	-0.153	-0.166

Table 5: Fairness Results for Different Models

7 Analysis of the Results and Concluding Remarks

The fairness analysis of the machine learning models, as presented in Tables 4 and 5, reveals important insights into the impact of these models on different demographic groups, specifically focusing on sex and race as protected attributes.

For sex-based fairness, the metrics evaluated indicate that all models exhibit some degree of bias against females. The **Statistical Parity Difference (SPD)** ranges from -0.112 for XGBoost to -0.178 for Logistic Regression, indicating that females are less likely to receive positive outcomes compared to males. The **Equal Opportunity Difference (EOD)**, which measures the true positive rate differences, ranges from -0.267 for XGBoost to -0.33 for both Linear SVC and Logistic Regression, highlighting a substantial disparity in model performance favoring males. Similarly, the **Average Odds Difference (AOD)**, which measures the combined differences in false positive and true positive rates, ranges from -0.14 for XGBoost to -0.206 for Logistic Regression, showing consistent bias against females across all models.

In terms of race-based fairness, the metrics demonstrate more variability and less consistent bias compared to sex. The **SPD** shows slight favoritism towards certain racial groups, with values ranging from 0.141 for Logistic Regression to 0.18 for both XGBoost and Random Forest, indicating that individuals from privileged racial groups are more likely to receive positive outcomes. The **EOD** ranges from 0.014 for both Linear SVC and Logistic Regression to 0.042 for Random Forest, showing minor differences in true positive rates between privileged and unprivileged racial groups. The **AOD** values range from 0.072 for Logistic Regression to 0.109 for both XGBoost and Random Forest, indicating a relatively lower bias in false positive and true positive rates compared to the sex-based analysis.

When considering overall fairness, aggregating sex and race metrics, the **SPD** is consistently negative, ranging from -0.064 for XGBoost to -0.119 for Logistic Regression, suggesting a general trend towards unfair outcomes for unprivileged groups. The **EOD** ranges from -0.23 for XGBoost to -0.293 for both Linear SVC and Logistic Regression, highlighting significant disparities in favorable outcomes. The **AOD** also shows a consistent negative bias, ranging from -0.108 for XGBoost to -0.166 for Logistic Regression, indicating persistent unfairness across the models.

In conclusion, the results of this analysis underline the critical need to address biases in machine learning models, particularly regarding sex. Although race-based disparities are less pronounced, they still warrant attention. The consistent negative biases revealed by the SPD, EOD, and AOD metrics across different models highlight the importance of integrating fairness-aware algorithms and techniques in model training and evaluation to mitigate such biases and ensure equitable outcomes for all demographic groups. These findings should prompt further investigation into the sources of bias and the development of targeted strategies to enhance model fairness, thereby contributing to more just and unbiased decision-making systems. **These findings show that training models on datasets in which biased Approximate Functional Dependencies hold may result in unfair models, suggesting the pursuit of research activities both in the Software Engineering and in the Data Profiling realm to address bias issues.**

References

[1] Philip Bohannon et al. “A Cost-Based Model and Effective Heuristic for Repairing Constraints by Value Modification”. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data.* 2007, pp. 143–154.

[2] Wenfei Fan. “Dependencies Revisited for Improving Data Quality”. In: *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems.* 2012, pp. 5–16.

[3] Ykä Huhtala et al. “TANE: An Efficient Algorithm for Discovering Functional and Approximate Dependencies”. In: *The Computer Journal* 42.2 (1999), pp. 100–111.

[4] Xifeng Yan, Hong Zhang, and ChengXiang Zhang. “Discovering Approximate Functional Dependencies for Data Cleaning and Integration”. In: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 2007, pp. 396–405.

[5] Ninareh Mehrabi et al. “A Survey on Bias and Fairness in Machine Learning”. In: *ACM Computing Surveys* 54.6 (2021), 115:1–115:35. DOI: 10.1145/3457607.

[6] Solon Barocas, Moritz Hardt, and Arvind Narayanan. “Fairness and Machine Learning: Limitations and Opportunities”. In: (2019). Book in progress. Available online at http://fairmlbook.org.

[7] Sahil Verma and Julia Rubin. “Fairness Definitions Explained”. In: *Proceedings of the 2018 ACM/IEEE International Workshop on Software Fairness (FairWare).* 2018, pp. 1–7. DOI: 10.1145/3194770.3194776.

[8] Sriram Vasudevan and Krishnaram Kenthapadi. “Lift: A scalable framework for measuring fairness in ml applications”. In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management.* 2020, pp. 2773–2780.

[9] Jie M Zhang and Mark Harman. ““Ignorance and Prejudice” in Software Fairness”. In: *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE).* IEEE. 2021, pp. 1436–1447.

[10] Joymallya Chakraborty, Suvodeep Majumder, and Tim Menzies. “Bias in machine learning software: why? how? what to do?” In: *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering.* 2021, pp. 429–440.

[11] Zafeiria Moumoulidou, Andrew McGregor, and Alexandra Meliou. “Diverse Data Selection under Fairness Constraints”. In: *arXiv preprint arXiv:2010.09141* (2020).

[12] Brianna Richardson and Juan E. Gilbert. “A Framework for Fairness: A Systematic Review of Existing Fair AI Solutions”. In: *CoRR* abs/2112.05700 (2021). arXiv: 2112.05700. URL: https://arxiv.org/abs/2112.05700.

[13] Simon Caton and Christian Haas. “Fairness in Machine Learning: A Survey”. In: *CoRR* abs/2010.04053 (2020). arXiv: 2010.04053. URL: https://arxiv.org/abs/2010.04053.

[14] Fabio Azzalini, Chiara Criscuolo, and Letizia Tanca. “E-FAIR-DB: Functional Dependencies to Discover Data Bias and Enhance Data Equity”. In: *J. Data and Information Quality* 14.4 (2022). ISSN: 1936-1955. DOI: `10.1145/3552433`. URL: `https://doi.org/10.1145/3552433`.

[15] Loredana Caruccio, Vincenzo Deufemia, and Giuseppe Polese. “Mining relaxed functional dependencies from data”. In: *Data Mining and Knowledge Discovery* 34 (Mar. 2020). DOI: `10.1007/s10618-019-00667-7`.