Automated Data Cleaning Can Hurt Fairness in Machine Learning-based Decision Making

Shubha Guha
s.guha@uva.nl
University of Amsterdam

Falaah Arif Khan
fa2161@nyu.edu
New York University

Julia Stoyanovich
stoyanovich@nyu.edu
New York University

Sebastian Schelter
s.schelter@uva.nl
University of Amsterdam

Abstract—**In this paper, we interrogate whether data quality issues track demographic characteristics such as sex, race and age, and whether automated data cleaning — of the kind commonly used in production ML systems — impacts the fairness of predictions made by these systems. To the best of our knowledge, the impact of data cleaning on fairness in downstream tasks has not been investigated in the literature.**

We first analyze the tuples flagged by common error detection strategies in five research datasets. We find that, while specific data quality issues, such as higher rates of missing values, are associated with membership in historically disadvantaged groups, poor data quality does not generally track demographic group membership. As a follow-up, we conduct a large-scale empirical study on the impact of automated data cleaning on fairness, involving more than 26,000 model evaluations on five datasets. We observe that, while automated data cleaning has an insignificant impact on both accuracy and fairness in the majority of cases, it is *more likely to worsen fairness than to improve it*, especially when the cleaning techniques are not carefully chosen. This finding is both significant and worrying, given that it potentially implicates many production ML systems. We make our code and experimental results publicly available.

The analysis we conducted in this paper is difficult, primarily because it requires that we think holistically about disparities in data quality, disparities in the effectiveness of data cleaning methods, and impacts of such disparities on ML model performance for different demographic groups. Such holistic analysis can and should be supported with the help of data engineering research. Towards this goal, we envision the development of fairness-aware data cleaning methods, and their integration into complex pipelines for ML-based decision making.

I. Introduction

Software systems that learn from user data with machine learning (ML) are in ubiquitous use in critical decision-making processes such as loan approvals, hiring, and prioritizing access to medical interventions. Unfortunately, if left unchecked, such applications often reproduce or even amplify pre-existing bias in the data, leading to unlawful discrimination [1].

Data quality and fairness in production ML. Most ML applications in production are data-intensive, and require data cleaning [2]. Such applications regularly acquire new training data in short intervals (e.g., nightly from log files), and subsequently retrain and redeploy models, which then make predictions on previously unseen data. Real-world data — processed by production ML systems — inevitably includes data errors [3]–[6]. Due to large data size and short redeployment intervals, data quality issues are often addressed with automated cleaning techniques (e.g., to impute missing values, which many ML models cannot handle directly).

There are indications that data from historically disadvantaged groups may be more likely to suffer from poor quality, such as higher occurrence of missing values [7]. Such "heteroskedastic noise" in the data, in turn, has the potential to negatively impact ML model fairness [8]. Yet, while there is plenty of evidence that data quality issues hurt the predictive accuracy of ML models [5], it is unclear whether (1) poor data quality tracks membership in disadvantaged groups, and (2) attempts to improve data quality through automated cleaning impact the *fairness* of ML models (e.g., by amplifying disparities in prediction quality among groups).

To the best of our knowledge, these questions have not been investigated in prior work. On the one hand, the growing body of work on joint cleaning and learning [5], [9]–[11] focuses on predictive accuracy but not on fairness. On the other hand, research on fairness in ML usually ignores data quality issues; it is common, for example, to simply remove tuples with missing values from the data before experimentation [12], [13]. Moreover, existing data-centric work on fairness either focuses on coverage (e.g., underrepresentation) at training time [8], [14], [15] (and not on repairing erroneous tuples), or it introduces synthetically-generated errors only [16]–[18], making it difficult to judge how representative the results are of real world settings.

Quantifying the impact of automated data cleaning on fair decision-making. What is the impact of data errors and automated cleaning on model performance, both over-all and for subsets of the data corresponding to different demographic groups? This question is both crucial and understudied, with very real implications for production ML systems currently used for critical decision-making. A major challenge is that there is no "clean" ground truth available for datasets that are commonly used for ML fairness research. Furthermore, such datasets are hard to clean manually, in part because validating data errors would require access and corroboration through secondary data sources (e.g., bank records or medical files), which raises substantial privacy and data protection concerns.

Therefore, instead of trying to quantify the quality of the data directly, we tailor our research questions to address the two common stages of automated data cleaning: (1) error detection, which flags potentially erroneous tuples, and (2) data repair, which attempts to correct the erroneous tuples:

- *RQ1. Does the incidence of data errors track demographic group membership in ML fairness datasets?*

- *RQ2. Do common automated data cleaning techniques impact the fairness of ML models trained on the cleaned datasets?*

To address *RQ1*, we analyze the tuples flagged by common error detection strategies in five widely used fairness benchmark datasets, with respect to groups based on sex, race, and age (Section III). To address *RQ2*, we conduct an empirical study of the impact of data cleaning on model fairness (Section IV), by applying common automated data cleaning techniques to the potentially erroneous tuples detected in *RQ1*. Our study involves training and evaluating more than 26,000 models and, in contrast to existing work, does not inject synthetic noise but works with the raw data as provided.

Contributions. We make the following contributions.

- We find that higher rates of missing values are associated with membership in historically disadvantaged groups. However, for other types of data errors, we do not find sufficient evidence that poor data quality tracks demographic group membership (Section III).
- We find that, while automated data cleaning has an insignificant impact on both accuracy and fairness in most cases, it is *more likely to worsen fairness than to improve it in cases where an impact is observed*, especially when the cleaning techniques are not carefully chosen. This finding is both significant and worrying, given that it potentially implicates many production ML systems! The observed effect varies based on dataset, fairness metric, and type of error being repaired. In many cases, we do not encounter a configuration that simultaneously improves both fairness and accuracy (Section IV).
- We outline which cleaning techniques, error detection strategies and ML models turned out to be most beneficial for fairness and accuracy in our study (Section IV).
- We discuss the implications of our findings, and outline research challenges and directions for follow-up work in Section V. Furthermore, we provide the code and results for our study and experiments for reproducibility and follow-up research.[1]

II. Preliminaries

We introduce sensitive demographic attributes, datasets, error detection strategies, as well as automated data cleaning and repair methods used in our study.

Sensitive attributes. We investigate disparities with respect to sensitive attributes based on which unlawful discrimination in decision-making has been observed [1], e.g., violating US labor law [19] or European non-discrimination law [20]. Given a sensitive attribute, we partition the data into tuples belonging to a *privileged group* and all other tuples as belonging to a *disadvantaged group*.

We consider sex (with 'male' as the privileged group), race (with 'white' as the privileged group) and age (with people older than 30 years as the privileged group). Note that which demographic group is considered privileged vs. disadvantaged is task-specific, and is designated as appropriate for the benchmark datasets and tasks described below. For example, older age is considered privileged in the context of lending, but disadvantaged in the context of hiring.

Benchmark datasets. We use five publicly available datasets listed in Table I from three source domains: census, finance, and healthcare. These datasets are commonly used in research on responsible machine learning and data management [7], [8], [12], [21]. Each dataset is associated with a binary classification task. In our setup, the positive class always corresponds to the desirable outcome for the individuals in the dataset, such as being considered creditworthy or being prioritized for access to healthcare resources. Note that the choice of sensitive attribute(s) is taken from existing research on these datasets [7], [8], [12], [21].

name	source	number of tuples	number of attributes	sensitive attribute(s)
`adult`	census	48,844	12	sex, race
`folk`	census	378,817	10	sex, race
`credit`	finance	150,000	8	age
`german`	finance	1,000	18	age
`heart`	healthcare	70,000	11	sex

TABLE I
Benchmark datasets used in ML fairness research.

The `adult`[2] dataset contains demographic and financial data, and the target variable denotes whether a person earns more than 50,000 dollars per year or not. This dataset has been used extensively to evaluate fairness in predictions of credit-worthiness. Recent work proposes to "retire" this dataset due to both unclear data origins and the apparent — and unrepresentative — class-label imbalance, which renders the prediction task unrealistic [21]. We include this dataset in our study as a way to complement these concerns from a data management perspective, exposing additional data quality issues. The `folk`[3] dataset is based on US census data and has been proposed as a replacement for the problematic `adult` [21] dataset, to be used for financial decisions. We use a subset of the data from the census in California in 2018, and replicate the prediction task from `adult`. The `credit`[4] and `german`[5] datasets contain financial information, and the target variable denotes whether a person has a good credit score. The `heart`[6] dataset consists of patient measurements with respect to cardiovascular diseases, and the target variable denotes the presence of a heart disease. This dataset has been used to evaluate fairness of predictive tasks that allocate access to priority medical care for individuals.

Error detection strategies. We apply common error detection strategies that have been proposed in the data cleaning literature [3], [22], [23] and are also used in studies about the impact of data cleaning on machine learning tasks [5].

[1] https://github.com/amsterdata/demodq

[2] https://archive.ics.uci.edu/ml/datasets/adult

[3] https://github.com/zykls/folktables

[4] https://www.kaggle.com/c/GiveMeSomeCredit

[5] https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)

[6] https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset

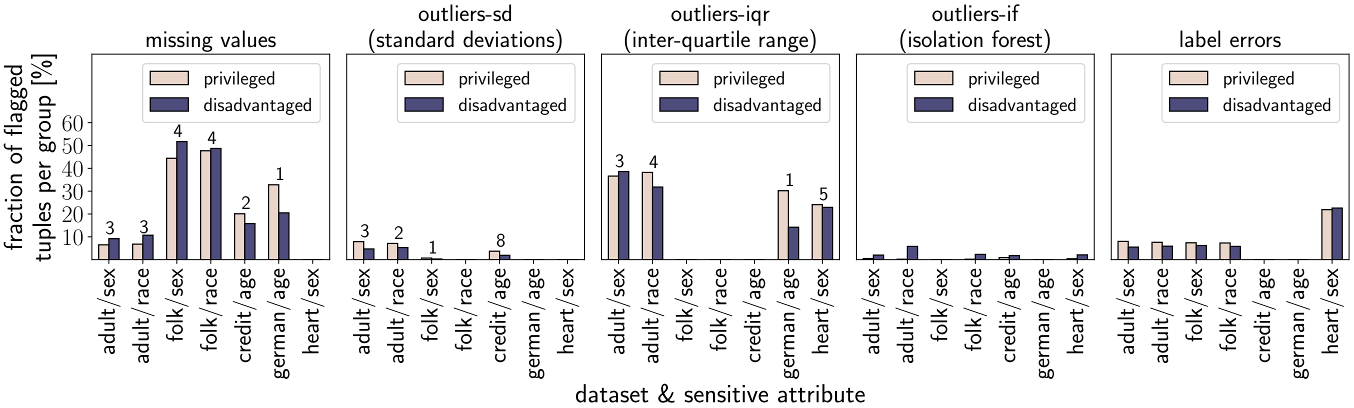

Fig. 1. Disparate proportions of tuples flagged by common error detection strategies for the privileged and disadvantaged groups. The numbers above the bars denote the number of attributes with a particular error type for the univariate error detection strategies which operate on the attribute-level (`missing values`, `outliers-sd`, `outliers-iqr`). While historically disadvantaged groups are subject to higher rates of missing values in the majority of cases, we do not find sufficient evidence of a demographic dependency in data errors.

Missing values. We identify tuples with `missing` values by detecting NULL and NaN values in the datasets.

Outliers. We detect numerical outliers with the following techniques: (i) `outliers-sd` – we consider a value of a column to be an outlier if it is more than n standard deviations away from the mean of the column (with $n = 3$); (ii) `outliers-iqr` – we consider a value of a column to be an outlier if it lies outside of the interval $[p_{25} - k \cdot \text{iqr}, p_{75} + k \cdot \text{iqr}]$ with $k = 1.5$. Note that iqr refers to the inter-quartile range defined as the difference between the 75th and 25th percentile of the column distribution: iqr $= p_{75} - p_{25}$; (iii) `outliers-if` – a tuple is considered to be an outlier if it is identified as such by an isolation forest trained on the data with a contamination parameter of 0.01. Note that `outliers-sd` and `outliers-iqr` are univariate techniques that inspect individual attributes, while the multivariate approach `outliers-if` inspects whole tuples.

Label errors. An ML-specific type of error are mislabeled examples: tuples with the wrong prediction label assigned to them. Such errors have recently received a lot of attention, due to the fact that they are pervasive in widely used benchmarking datasets for ML [24]. We detect tuples with potential `label errors` tuples with the cleanlab [25] library, using a logistic regression model as the base classifier. Cleanlab identifies label errors in datasets by estimating the joint distribution between noisy (given) labels and uncorrupted (unknown) labels.

Limitations. Unfortunately, there are no known integrity constraints available for the datasets (e.g., in the form of functional dependencies or denial constraints [26]) and no verified sets of clean records, which prevents us from applying more advanced cleaning and error detection techniques such as HoloClean [27], HoloDetect [28] or kNN-Shapley [29]. We consider it an interesting avenue for future work to include these approaches on appropriate datasets and tasks.

Automated repair methods. We apply standard techniques for repairing erroneous tuples, which are implemented in popular data science packages such as scikit-learn[7] or pandas, and used in existing studies on joint cleaning and learning [5]. We apply several methods to impute missing values, namely, via the column mean or mode for numerical columns, and via the mode or a constant "dummy" value for categorical columns. We repair outlier values in numerical columns by replacing them with the mean or mode of the column. We repair label errors by flipping the labels of flagged tuples.

III. Indications of Demographically Disparate Data Quality Issues

To address *RQ1*, we search for cases in which the error detection strategies flag significantly different fractions of the privileged and disadvantaged groups, based on sex, race or age. For a dataset D, let the Boolean predicate priv(t) evaluate if tuple $t \in D$ belongs to the privileged group. Further, let the Boolean error function $\sigma(t)$ evaluate if t is considered erroneous by detection strategy σ.

To identify statistically significant disparities, we compute the number of erroneous tuples $|\{t \in D \,|\, \text{priv}(t) \wedge \sigma(t)\}|$ from the privileged group, the number of erroneous tuples $|\{t \in D \,|\, \neg\text{priv}(t) \wedge \sigma(t)\}|$ from the disadvantaged group, and conduct a G^2 significance test with a threshold of $p = .05$. We report only cases that pass this test. We run the error detectors for disparities w.r.t. sex on the `adult`, `folk`, and `heart` datasets, w.r.t. race on the `adult` and `folk` datasets, and w.r.t. age on the `credit` and `german` datasets. We plot the results in Figure 1, distinguished by dataset, sensitive attribute, error type and detection strategy. The bars represent the fraction of tuples from the privileged group and from the disadvantaged group that were flagged as erroneous. The

[7] https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html

numbers above the bars denote the number of attributes with a particular error type for the univariate error detection strategies which operate on the attribute-level (missing values, outliers-sd, and outliers-iqr).

Results. We find that all three data errors (missing values, outliers and label noise) are frequently detected in the research datasets.[8] These errors are flagged in disparate proportions for different datasets and protected characteristics, and, strikingly, error detection strategies often identify large fractions of erroneous tuples (e.g., up to 51% of the tuples of a particular group). Notably, adult — one of the most widely used datasets in fair ML — is the only dataset for which all six error detectors flag tuples with significant disparities, for both sex and race. We interpret this as additional evidence that it is time to "retire" adult [21].

Disparities in missing values. We find that tuples from the disadvantaged group are subject to missing data more frequently: in four out of six dataset/sensitive attribute combinations, and in 14 out of 17 attributes with disparate proportions of missing data, the fraction of tuples from the disadvantaged group is higher than the fraction of tuples from the privileged group.

Disparities in outliers. We see a mixed picture w.r.t. outliers, where it varies strongly which group is more affected. Additionally, there are several cases where we encounter disparate proportions of outliers with only a particular detection technique but not with others. Additionally, we find that the amount of outliers detected heavily varies based on the applied detection strategy.

Disparities in predicted label errors. For label errors, we find that, in the majority of cases (4 out of 6), the fraction of tuples from the privileged group in the mislabeled data is higher than the fraction of tuples from the disadvantaged group. (Note that these labeling errors are predicted, and that we do not have access to the ground truth.) We drill in on the type of label error — false positive or false negative — and find no significant differences between the privileged and the disadvantaged groups in most cases. However, in one case (in the heart dataset) the fraction of false positives was significantly higher for the privileged group than for the disadvantaged (57.7% vs. 52.2%, respectively), and the trend was reversed for the false negatives (42% vs. 47.8%, respectively). This is potentially problematic, because false positives can amplify the advantage, while false negatives can exacerbate the disadvantage for the respective groups.

Discussion. Overall, while we do find strong indication of a large number of data quality issues in benchmark datasets, we do not find sufficient evidence that these potential data errors track demographic group membership with respect to sex, race and age. In datasets such as folk and heart, overall, errors are detected more frequently in the disadvantaged group, but the disparity in errors between groups is small. In datasets such as credit and german, where the disparity in the incidence of errors across groups is large, errors do not systematically occur more frequently for the disadvantaged group: in 4 out of 6 configurations, the fraction of errors in the privileged group is higher than in the disadvantaged group.

The results of *RQ1* are counter-intuitive to the hypothesis that data from historically marginalized groups is more likely to be erroneous, and motivates our large-scale empirical study for a principled answer to *RQ2*. We discuss further implications of this finding in Section V.

IV. Impact of Automated Data Cleaning on Fairness

In the following, we address *RQ2* and study the impact of applying data repairs to the flagged tuples as part of the training and evaluation of machine learning models for decision-making.

Setup. Our goal is to quantify the downstream impact of automated data repairs on the *fairness and accuracy* of ML models. We adapt the existing CleanML benchmark for joint data cleaning and model training [5], and compute several fairness metrics during evaluation. For that, we integrate our five datasets into CleanML, and extend its code to record group memberships per tuple and to compute group-specific confusion matrices on the test set.

Classification models and training procedure. We use three different ML models, each of which we tune using 5-fold cross-validation: logistic-regression (log-reg) with a tuned learning rate, nearest neighbors (knn) with a tuned number of neighbors, and gradient-boosted decision trees (xgboost) with a tuned maximum tree depth. During each run, we sample 15,000 records from a given dataset, randomly split these into train and test set, and evaluate five different model instances (with different random seeds for the hyperparameter search) per split. We repeat this 20 times per configuration (dataset/-model/error/repair), resulting in the training and evaluation of 100 models per configuration.

Evaluation. For each run, we evaluate the predictions of the corresponding model (learned on the repaired train set) on an equivalently repaired test set. We compare these predictions to the "dirty" baseline predictions of a model, trained and evaluated on the "dirty" version of the data. For each prediction, we compute the following two fairness metrics:

- *Predictive parity* is satisfied if a classifier has equal precision for the subjects in the privileged and disadvantaged groups. This metric is computed as $\frac{TP_{\text{priv}}}{TP_{\text{priv}}+FP_{\text{priv}}} - \frac{TP_{\text{dis}}}{TP_{\text{dis}}+FP_{\text{dis}}}$, and denotes equal probability of a correct positive prediction for the groups.
- *Equal opportunity* is satisfied if a classifier has equal recall for the subjects in the privileged and disadvantaged groups. This metric is computed as $\frac{TP_{\text{priv}}}{TP_{\text{priv}}+FN_{\text{priv}}} - \frac{TP_{\text{dis}}}{TP_{\text{dis}}+FN_{\text{dis}}}$. Note that only records with a positive label influence this metric.

In line with existing research [13], we choose these two metrics because they intuitively represent the opposing interests of two key stakeholders in many decision making processes — individuals who seek access to resources, and vendors who grant access. For example, in lending, the bank, on the

[8] Note that the heart dataset has no missing values at all.

one hand, wants high precision (to avoid giving loans to creditors who might not have the means to repay them), while customers, on the other hand, want high recall (to avoid being denied a loan that they would have been able to repay).

Error detection and repairs. We detect errors and repair flagged tuples as outlined in Section II. We select different variants of missing value imputation as outlined earlier. Note that most classifiers cannot naturally handle missing values, which requires us to define a modified version of the data as the 'dirty' version. For the 'dirty' setup, we remove tuples with missing values from the training data and impute them with the mean for numerical columns and dummy for categorical columns on the test data. Note that one cannot simply remove tuples with missing values from the data during prediction in a real-world setup, therefore we have to impute on the test set as well for consistency. For other types of errors, missing values have to be removed from the data beforehand. We detect outliers and impute them as outlined earlier in Section II. In the "dirty" setup, we simply retain the outliers in both the train set and the test set. For labeling errors, we run cleanlab for detection and flip the labels of identified tuples as a repair technique. For the "dirty" setup, we leave the labels as is in both train and test set. Note that we never flip labels on the test set, as this would make the prediction results incomparable with the other experiments.

Results and discussion. We evaluate 26,400 models in total, and compute a result table from our experiments, where each row contains the result of a particular configuration with respect to a dataset, sensitive attribute, fairness metric, model, error type, detection method, repair method, and indicators for the impact on fairness and accuracy.[9] The impact on fairness as well as the impact on accuracy of a configuration can be positive, negative or insignificant. We determine this by comparing the resulting 100 fairness and accuracy scores from the "dirty" baseline (with no cleaning) to the scores from a cleaning configuration (dataset, sensitive attribute, fairness metric, error, detection, repair). We leverage a sequence of paired sample t-tests as proposed by CleanML [5] with a threshold for the p-value of .05 adjusted by Bonferroni correction to account for multiple hypothesis tests. In the following, we analyze this result table. Tables II, III & IV report the impact of auto-cleaning on fairness and accuracy for missing values, outliers and label errors (note that the counts in the cells denote the number of configurations, and that each configuration represents 100 model evaluations).

Impact of repairing missing values. In general, we find that auto-cleaning missing values does not degrade downstream model performance: 38% of the times it has no significant impact on accuracy, and nearly half of the times (49.1%) it results in an improvement. Similarly, 59.3% of the times it has no significant impact on fairness. Worryingly however, when cleaning does have an impact on fairness, it is more likely to have an adverse impact (23.6%) than a positive one (17.1%).

			accuracy		
		worse	insignificant	better	
fair.	worse	1.9% (4)	10.6% (23)	11.1% (24)	23.6% (51)
	insign.	9.7% (21)	25.5% (55)	24.1% (52)	59.3% (128)
	better	1.4% (3)	1.9% (4)	13.9% (30)	17.1% (37)
		13.0% (28)	38.0% (82)	49.1% (106)	

TABLE II
IMPACT OF AUTO-CLEANING MISSING VALUES.

Impact of repairing outliers. We see a similar trend for outlier-repair: Most of the times, cleaning has an insignificant impact on both accuracy (49.7%) and fairness (72.2%), and, on balance, when cleaning does have an impact on fairness, it is more likely to worsen (19.6%) than improve it (8.2%). Interestingly, however, we notice that fairness-gains track accuracy-gains: when cleaning improves accuracy, it is also more likely to improve fairness (6.1%) than worsen it (4.0%). And, when it worsens accuracy, it is also much more likely to worsen fairness (9%) than improve it (1.9%).

			accuracy		
		worse	insignificant	better	
fair.	worse	9.0% (34)	4.5% (17)	6.1% (23)	19.6% (74)
	insign.	11.4% (43)	42.9% (162)	18.0% (68)	72.2% (273)
	better	1.9% (7)	2.4% (9)	4.0% (15)	8.2% (31)
		22.2% (84)	49.7% (188)	28.0% (106)	

TABLE III
IMPACT OF AUTO-CLEANING OUTLIERS.

Impact of repairing predicted label errors. For label errors, the accuracy impact is extremely strong: auto-repairing label errors does not hurt accuracy in any configuration, and it improves accuracy over 90% of the time. Interestingly, while improving accuracy, cleaning is (almost) equally likely to improve (23.8%), worsen (31%), or have no impact on fairness (35.7%).

			accuracy		
		worse	insignificant	better	
fair.	worse	0.0% (0)	2.4% (1)	31.0% (13)	33.3% (14)
	insign.	0.0% (0)	7.1% (3)	35.7% (15)	42.9% (18)
	better	0.0% (0)	0.0% (0)	23.8% (10)	23.8% (10)
		0.0% (0)	9.5% (4)	90.5% (38)	

TABLE IV
IMPACT OF AUTO-CLEANING LABEL ERRORS.

In general, we find that fairness is not significantly impacted in the majority of cases (42.9% to 72.2%). However, *across all types of errors, automated cleaning is more likely to worsen than to improve fairness*! This finding illustrates that we should be concerned about the choice of cleaning techniques for models in production and that we need to develop means to carefully choose appropriate cleaning techniques to prevent a negative impact on fairness. When we additionally calculate the impact dependent on the fairness metric, we find a large negative imbalance for equal opportunity (worsening of fairness in 32.7% compared to positive impact in 7.5% of cases) and a slightly positive balance for predictive parity (worsening of fairness in 11% compared to positive impact in 16.9% of cases). This result motivates us to look at the impact of automated cleaning on a more granular level.

[9] https://github.com/amsterdata/demodq/blob/master/cleanml.csv

For which cases (dataset, error and fairness metric) is cleaning potentially beneficial at all? In order to assess whether it would be possible to carefully choose a beneficial cleaning technique for a given setting, we analyze for which of the cases in our study we encounter a beneficial auto-cleaning technique at all. We define a case as a combination of a fairness metric (predictive parity or equal opportunity), a dataset with a sensitive attribute, and an error type (missing values, outliers or label errors), resulting in 40 different cases in total.

A promising finding is that for nearly every case (38 out of 40), we encounter at least one cleaning technique which does not worsen fairness. In half of the cases (20 out of 40), there exists a cleaning technique which improves fairness, while we can improve both fairness and accuracy simultaneously only in 15 out of 40 cases. When taking a deeper look, we find stark differences with respect to the fairness metrics (for each of which we have 20 cases). Auto-cleaning is more likely to improve predictive parity than equal opportunity: In 14 out of 20 cases, cleaning improves the predictive parity fairness metric, compared to only 6 cases for equal opportunity. In 13 cases, cleaning improves both accuracy and fairness for predictive parity, compared to only 2 for equal opportunity. We attribute this to the fact that equal opportunity is only influenced by predictions on the smaller fraction of tuples with the desirable positive label. From these results, we conclude that the impact and benefits of auto-cleaning on fairness and accuracy heavily depends on the choice of fairness metric, and must therefore be carefully evaluated in practice!

Which repair and detection techniques produce the most gains? Next, we focus on configurations with a positive impact on fairness, and analyze the applied detection and repair techniques in such cases.

For missing values, we do not encounter a dominating imputation approach for numerical columns. However, for categorical columns, "dummy" imputation with a constant value turns out to be most beneficial for fairness (with fairness improvements in 22 cases, compared to only 15 cases with a different imputation technique). We attribute this to the fact that dummy imputation allows the model to identify tuples with missing values and learn extra parameters for them (which is not the case for mode and mean imputation). For example, in the `folk` dataset, the accompanying datasheet makes it clear that missing values are typically 'Not Applicable (N/A)', based on values in another column; e.g., Occupation (OCCP) and Class of Worker (COW) are missing for people with Age (AGEP) less than 18. In this case, the missing value is actually a special N/A value, and dummy imputation allows the model to learn such a dependency.

For outliers, we observe no noticeable differences between the repair techniques. However, we find a clear difference when analyzing the detection techniques. In 16 cases, cleaning the outliers detected with the standard deviation rule (`outliers-sd`) increases the fairness of the resulting model, compared only 11 cases for detection with an isolation forest (`outliers-if`) and 4 cases for detection with the interquartile range rule (`outliers-iqr`). In combination with the fractions of flagged tuples from Figure 1, these findings uncover the efficacy of different detection techniques: `outliers-iqr` flags a large amount of records, but apparently catches less impactful examples, and demonstrates a high incidence of false positives. Conversely, `outliers-if` shows more false negatives and flags too few records to have a positive downstream effect. `outliers-sd` seems to have best performance in optimally flagging data outliers.

	auto-cleaning makes		
model	**fairness worse**	**fairness better**	**fairness & accuracy better**
`xgboost`	21.2% (45)	10.8% (23)	6.6% (14)
`knn`	24.5% (52)	13.7% (29)	11.8% (25)
`log-reg`	19.8% (42)	12.3% (26)	7.5% (16)

TABLE V
IMPACT OF AUTO-CLEANING ON ACCURACY AND FAIRNESS FOR DIFFERENT ML MODELS ON 212 CONFIGURATIONS IN TOTAL. WE LIST CASES WHERE FAIRNESS GETS WORSE, FAIRNESS GETS BETTER, AND WHERE BOTH FAIRNESS AND ACCURACY GET BETTER. AUTO-CLEANING IS MORE LIKELY TO WORSEN THAN TO IMPROVE FAIRNESS ACROSS ALL MODELS.

Model choice. We also investigate the influence of the choice of ML model on the impact on fairness and accuracy. The highest accuracy over all tasks is provided by the gradient boosted trees technique `xgboost`. Apart from that, we find that all models perform comparably with respect to the impact of auto-cleaning on the fairness of their predictions (Table V). In the majority of cases, this impact is insignificant, however, if there is an impact, auto-cleaning is more likely to worsen (19.8% to 24.5% of the cases) than to improve fairness (10.8% to 12.3% of the cases).

Logistic regression (`log-reg`) turns out to be the "safest" choice in our study, with the smallest fraction of cases with negative impact (19.8%), while `xgboost` benefits least from cleaning (fairness and accuracy improve in only 6.6% of cases). Still, `xgboost` would be the model of choice in general, as it had the highest accuracy in light of missing values and outliers, while `log-reg` only had the highest accuracy in cases with cleaned label errors. We find that `knn` benefits most from cleaning, but does not outperform the other models in terms of accuracy in any configuration.

V. VISION: FAIRNESS-AWARE DATA CLEANING

The analysis we conducted in this paper is difficult, primarily because it requires that we think holistically about disparities in data quality, disparities in the effectiveness of data cleaning methods, and impacts of such disparities on ML model performance for different demographic groups. Such holistic analysis can and should be supported by data engineering tools, but it requires substantial future research. To detect disparities in data quality, and mitigate the impact of such disparities on the performance of ML models downstream, we envision the development of fairness-aware data cleaning methods and their integration into complex data-intensive pipelines.

Implications for ML in production. While we did notice that historically disadvantaged groups are subject to higher rates of missing values in the majority of cases, we did not find sufficient evidence of a demographic dependency in data errors. This is counter-intuitive to a socio-technical framing, which posits that marginalized groups also appear noisier in the data (have more data errors), and could embolden data scientists to not worry about disparate effects along demographic lines when applying automated cleaning procedures.

However, our second result about the downstream effect of automated cleaning demonstrates that repairing data errors does, in fact, distribute gains disparately across demographic groups! In Section IV, we found that automated data cleaning can have a negative impact on fairness, and was, in our study, more likely to worsen fairness than to improve it. This is extremely worrying, due to the potential negative impact on the fairness of decisions made by many ML systems that are already in production.

The good news is, however, that we encountered at least one configuration for almost every case (dataset, error type, cleaning method, fairness metric) that did not negatively impact the fairness of model predictions. This indicates that we can — and should — mitigate any potential negative impact of automated cleaning with the help of a principled methodology for selecting an appropriate cleaning procedure. Our results underscore the importance of such a methodology, and motivate its development.

Open questions and research directions. Our findings indicate that we are either unable to detect demographically-salient data errors with current approaches, or that current cleaning procedures are not equally 'effective' for different demographic groups, or — most disturbingly — we are seeing failure modes in both detection and repair. In order to confirm whether the disparate proportions of tuples flagged by the error detection strategies in Section III correspond to actual errors, one would need to repeat this analysis on a dirty fairness-critical dataset where the clean ground truth is available. Thus future work on fairness-aware data cleaning must include additional empirical evaluation.

Our findings from the study in Section IV impose the immediate question of how to choose a cleaning technique that does not negatively impact fairness. Additionally, it would be interesting to analyze whether more advanced cleaning/error detection techniques [11], [27], [28] are impacted in a similar way (note that we had to exclude them from our study due to a lack of clean example tuples and integrity constraints). An important long-term research question in fairness-aware data cleaning is whether it will be sufficient to appropriately choose from existing cleaning techniques or whether we would need new fairness-aware cleaning procedures. The selection of cleaning techniques and model hyperparameters is typically steered by cross-validation techniques which aim for the highest accuracy. A promising direction here might be to extend existing techniques and implementations to adhere to fairness constraints during the selection procedure. A starting point for designing new cleaning techniques is the identification of input tuples with negative impact on fairness, which would then need to be cleaned in a fairness-enhancing manner. Several techniques for identifying such tuples have recently been proposed, e.g., by computing Shapley values with respect to a given fairness metric [30] or via causal explanations [13].

Finally, two limitations of our study are that we did not consider intersectional formulations of demographic characteristics, and mainly worked with US-centric datasets (which are common in fairness research). These limitations should be overcome in future work on fairness-aware data cleaning.

Acknowledgements. *This work was supported by in part by Ahold Delhaize, as well as by the NSF Awards No. 1922658 and 1916505, and by UL Research Institutes through the Center for Advancing Safety of Machine Intelligence. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.*

References

[1] J. Stoyanovich, S. Abiteboul, B. Howe, H. Jagadish, and S. Schelter, "Responsible data management," *Communications of the ACM*, 2020.

[2] N. Polyzotis, S. Roy, S. E. Whang, and M. Zinkevich, "Data lifecycle challenges in production machine learning: a survey," *SIGMOD Record*, vol. 47, no. 2, 2018.

[3] J. M. Hellerstein, "Quantitative data cleaning for large databases," *UNECE*, vol. 25, 2008.

[4] I. F. Ilyas and F. Naumann, "Data errors: Symptoms, causes and origins," *IEEE Data Eng. Bull.*, p. 4, 2022.

[5] P. Li, X. Rao, J. Blase, Y. Zhang, X. Chu, and C. Zhang, "Cleanml: A benchmark for joint data cleaning and machine learning," *ICDE*, 2019.

[6] I. F. Ilyas and T. Rekatsinas, "Machine learning and data cleaning: Which serves the other?" *JDIQ*, 2020.

[7] S. Schelter, Y. He, J. Khilnani, and J. Stoyanovich, "Fairprep: Promoting data to a first-class citizen in studies on fairness-enhancing interventions," *EDBT*, 2019.

[8] I. Chen, F. D. Johansson, and D. Sontag, "Why is my classifier discriminatory?" *NeurIPS*, 2018.

[9] F. Neutatz, B. Chen, Z. Abedjan, and E. Wu, "From cleaning before ml to cleaning for ml." *IEEE Data Eng. Bull.*, vol. 44, no. 1, 2021.

[10] S. Krishnan, J. Wang, E. Wu, M. J. Franklin, and K. Goldberg, "Activeclean: Interactive data cleaning for statistical modeling," *PVLDB*, vol. 9, no. 12, 2016.

[11] B. Karlaš, P. Li, R. Wu, N. M. Gürel, X. Chu, W. Wu, and C. Zhang, "Nearest neighbor classifiers over incomplete information: From certain answers to certain predictions," *VLDB*, 2020.

[12] S. A. Friedler, C. Scheidegger, S. Venkatasubramanian, S. Choudhary, E. P. Hamilton, and D. Roth, "A comparative study of fairness-enhancing interventions in machine learning," in *FACT**, 2019.

[13] R. Pradhan, J. Zhu, B. Glavic, and B. Salimi, "Interpretable data-based explanations for fairness debugging," *SIGMOD*, 2022.

[14] A. Asudeh, Z. Jin, and H. Jagadish, "Assessing and remedying coverage for a given dataset," *ICDE*, 2019.

[15] Y. Roh, K. Lee, S. Whang, and C. Suh, "Sample selection for fair and robust training," *NeurIPS*, 2021.

[16] M. T. Islam, A. Fariha, A. Meliou, and B. Salimi, "Through the data management lens: Experimental analysis and evaluation of fair classification," *SIGMOD*, 2022.

[17] Z. Liu, Z. Zhou, and T. Rekatsinas, "Picket: Self-supervised data diagnostics for ml pipelines," *VLDBJ*, 2020.

[18] S. Caton, S. Malisetty, and C. Haas, "Impact of imputation strategies on fairness in machine learning," *Journal of Artificial Intelligence Research*, vol. 74, 2022.

[19] Federal Trade Commission, "Protections Against Discrimination and Other Prohibited Practices," https://www.ftc.gov/policy-notices/no-fear-act/protections-against-discrimination.

[20] European Commission, "What the European Commission is doing to protect your rights," https://ec.europa.eu/info/aid-development-cooperation-fundamental-rights/your-rights-eu/know-your-rights/equality/non-discrimination_en#what-the-european-commission-is-doing-to-protect-your-rights.

[21] F. Ding, M. Hardt, J. Miller, and L. Schmidt, "Retiring adult: New datasets for fair machine learning," *NeurIPS*, 2021.

[22] Z. Abedjan, X. Chu, D. Deng, R. C. Fernandez, I. F. Ilyas, M. Ouzzani, P. Papotti, M. Stonebraker, and N. Tang, "Detecting data errors: Where are we and what needs to be done?" *PVLDB*, vol. 9, no. 12, 2016.

[23] M. Mahdavi, Z. Abedjan, R. Castro Fernandez, S. Madden, M. Ouzzani, M. Stonebraker, and N. Tang, "Raha: A configuration-free error detection system," *SIGMOD*, 2019.

[24] C. G. Northcutt, A. Athalye, and J. Mueller, "Pervasive label errors in test sets destabilize machine learning benchmarks," *NeurIPS*, 2021.

[25] C. G. Northcutt, T. Wu, and I. L. Chuang, "Learning with confident examples: Rank pruning for robust classification with noisy labels," *UAI*, 2017.

[26] X. Chu, I. F. Ilyas, and P. Papotti, "Discovering denial constraints," *PVLDB*, vol. 6, no. 13, 2013.

[27] T. Rekatsinas, X. Chu, I. F. Ilyas, and C. Ré, "Holoclean: Holistic data repairs with probabilistic inference," *VLDB*, 2017.

[28] A. Heidari, J. McGrath, I. F. Ilyas, and T. Rekatsinas, "Holodetect: Few-shot learning for error detection," *SIGMOD*, 2019.

[29] R. Jia, D. Dao, B. Wang, F. A. Hubis, N. M. Gurel, B. Li, C. Zhang, C. J. Spanos, and D. Song, "Efficient task-specific data valuation for nearest neighbor algorithms," *VLDB*, 2019.

[30] B. Karlaš, D. Dao, M. Interlandi, B. Li, S. Schelter, W. Wu, and C. Zhang, "Data debugging with shapley importance over end-to-end machine learning pipelines," 2022. [Online]. Available: https://arxiv.org/abs/2204.11131