American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN: 2320-0936
Volume-15, Issue-5, pp-60-68
www.ajer.org
Research Paper Open Access

Evaluating the Robustness of Fair Machine Learning Under Noisy Protected Attributes

Nikoloz Svanidze
[1]PhD student, Faculty of Informatics and Control Systems, Georgian Technical University

ABSTRACT : *Fairness-aware machine learning methods often depend on protected attributes for assessment or mitigation, yet those attributes may be noisy or missing in operational data pipelines. This study evaluates that reliability risk for threshold-based fairness post-processing. Using the UCI Adult/Census Income dataset, Logistic Regression and Random Forest classifiers were trained to predict whether annual income exceeds 50K. The protected attribute sex was removed from the model input features and retained separately for mitigation and evaluation; race was also removed from the input feature matrix. Fairlearn Threshold Optimizer was applied with demographic parity and equalized odds constraints after a stratified 60/20/20 train, calibration, and test split across five random seeds. Protected-label flipping was simulated from 0.00 to 0.50, and missingness was represented as Unknown from 0.00 to 0.30. Final fairness metrics were always computed with the clean protected attribute, even when noisy or missing attributes were supplied to the post-processor. In the executed pipeline, unmitigated Logistic Regression had mean accuracy 0.851 and demographic parity difference 0.175. Clean demographic-parity post-processing reduced the demographic parity difference to 0.008 for Logistic Regression and 0.007 for Random Forest, but the average threshold-optimized demographic parity difference increased from 0.052 under clean attributes to 0.185 at 0.50 calibration-and-deployment noise and 0.081 at 0.30 missingness. The results suggest that protected-attribute quality is part of the reliability boundary of fairness controls.*

KEYWORDS: *machine learning fairness; protected attributes; threshold optimization; Adult dataset; responsible artificial intelligence; classification*

Date of Submission: 05-05-2026 Date of acceptance: 16-05-2026

I. INTRODUCTION

Machine learning models are increasingly embedded in decision-support systems for screening, ranking, triage, and eligibility assessment. In such settings, predictive performance is not the only engineering requirement. Developers and system owners must also evaluate whether model behavior differs across socially meaningful groups and whether any mitigation method can be maintained reliably after deployment. The fairness literature has shown that formal criteria such as demographic parity, equalized odds, and individual fairness capture different aspects of these concerns rather than a single universal definition of fairness [1]-[3].

Many group-fairness methods depend on protected attributes during model evaluation, mitigation, or deployment. This dependency is often treated as a modeling detail, but it is also a data-pipeline dependency. In real systems, protected attributes can be missing, self-reported inconsistently, linked across systems with errors, inferred from proxies, or unavailable at prediction time because of policy or privacy constraints. When a mitigation method depends on those labels, protected-attribute quality becomes part of the reliability boundary of the fairness-control system.

Threshold-based post-processing is especially useful for this stress test because it operates after a base model has been trained. Fairlearn Threshold Optimizer learns group-specific thresholding rules from calibration data and applies them during prediction [9]. If the protected labels used during calibration are wrong, thresholds can be learned for the wrong groups. If the protected labels used during deployment are wrong, the learned thresholds can be applied to the wrong individuals. This study therefore examines a practical engineering question rather than proposing a new fairness algorithm.

The study addresses three research questions. RQ1 asks how binary protected-label noise affects threshold-based fairness post-processing. RQ2 asks how missing protected-attribute information affects fairness

and predictive performance. RQ3 asks whether the observed effects are consistent across Logistic Regression and Random Forest base classifiers.

The contribution is empirical and engineering-oriented: a compact, reproducible stress-test pipeline for fairness post-processing under protected-attribute corruption; controlled simulations of label flipping and missingness for the protected attribute; comparative evaluation of unmitigated classifiers and Fairlearn ThresholdOptimizer under demographic parity and equalized odds; and an engineering interpretation of protected-attribute quality as a necessary part of fairness mitigation reliability.

II. RELATED WORK

Fairness in machine learning is a broad sociotechnical topic. Dwork et al. formalized individual fairness around the idea that similar individuals should be treated similarly [1]. Group criteria such as demographic parity compare selection rates across groups, while separation criteria such as equalized odds compare error behavior conditional on the true label [2]. Barocas, Hardt, and Narayanan emphasize that these criteria are useful abstractions but cannot resolve all legal, ethical, and social questions raised by automated decision-making [8].

Post-processing methods modify predictions or thresholds after a model has been trained. Hardt, Price, and Srebro introduced a threshold-based framework for equality of opportunity and related error-rate constraints [2]. Pleiss et al. later examined the tension between calibration and fairness constraints, showing that different fairness and score-quality goals can be incompatible in general [3]. Agarwal et al. proposed a reductions approach for fair classification, illustrating the broader family of optimization-based mitigation methods [4].

Fairness toolkits help translate research algorithms into applied workflows. Fairlearn provides metrics, visualization support, and mitigation algorithms including Threshold Optimizer [9]. AI Fairness 360 similarly provides a toolkit of metrics and mitigation algorithms for algorithmic bias assessment [10]. These packages make applied experiments reproducible, but they do not remove the need to validate input data quality and deployment assumptions.

Recent work directly studies noisy, uncertain, or unavailable protected attributes. Celis et al. developed a fair-classification framework with provable guarantees under noisy protected attributes [5]. Ghosh, Kvitca, and Wilson compared attribute-reliant, noise-tolerant, and attribute-blind classifiers under protected-attribute noise [6]. Shah et al. analyzed group fairness when sensitive attributes are uncertain and showed that enforcing fairness on uncertain sensitive attributes can fall short of the fairness attainable with clean attributes [7]. This manuscript complements those studies by focusing on a small, reproducible engineering stress test of threshold-based post-processing.

The Adult/Census Income dataset remains a common benchmark for fairness experiments because it is public, tabular, and includes demographic attributes used in many examples [11]. Its age and social context limit external validity, but those same properties make it useful for reproducible method comparison. The present study uses scikit-learn pipelines for preprocessing and base modeling [12], and all manuscript tables are regenerated from saved seed-level metrics rather than manually edited.

III. MATERIALS AND METHODS

Dataset

The UCI Adult/Census Income dataset was loaded with ucimlrepo.fetch_ucirepo(id=2). The cleaned file contains 48,842 records, 14 original input features before target separation, and a binary target indicating income greater than 50K versus income less than or equal to 50K [11]. The target distribution in the cleaned data is 37,155 records for income less than or equal to 50K and 11,687 records for income greater than 50K. The sex attribute contains 16,192 Female records and 32,650 Male records. The original features include 6 numeric columns and 8 categorical columns before the target is removed; after removing sex and race from the model input, 12 raw predictors remain before one-hot encoding.

The primary protected attribute is sex. The dataset attribute is used as recorded in adult and should not be interpreted as a comprehensive representation of gender identity in real-world systems. Sex and race were removed from the model feature matrix. Sex was retained separately for Fairlearn mitigation and fairness evaluation. Missing values were present in workclass (2,799), occupation (2,809), and native_country (857).

Preprocessing

Question marks and blank strings were treated as missing values. Categorical variables were imputed with the most frequent category and one-hot encoded with unknown categories ignored at transform time. Numeric variables were imputed with the median. Logistic Regression additionally used standard scaling for numeric variables. The implementation used scikit-learn Pipeline and Column Transformer objects so that imputation, scaling, and encoding were learned only from the training split. Although sex and race were removed as direct model inputs, variables such as marital status, relationship, education, occupation, and native country can still act as proxy variables; this is treated as a limitation of the stress-test design rather than removed by assumption.

Base Models

Two base classifiers were used. Logistic Regression used the lbfgs solver, maximum iterations of 1000, default L2 regularization with C = 1.0, class_weight = None, and the experiment seed as random_state. Random Forest used 150 trees, min_samples_leaf = 5, max_depth = None, min_samples_split = 2, class_weight = None, n_jobs = -1, and the experiment seed as random_state. These values were verified from the project source code.

Fairness Post-Processing Method

Fairlearn Threshold Optimizer was used as the post-processing mitigation method. The base estimator was trained first and supplied to Threshold Optimizer with prefit = True. The post-processor used predict_proba scores, objective = accuracy_score, grid_size = 100, and constraints equal to either demographic parity or equalized odds. Demographic parity targets selection-rate parity across protected groups. Equalized odds targets parity in both true positive rates and false positive rates. Threshold optimization was fitted on calibration data and evaluated on held-out test data.

Protected-Attribute Noise and Missingness

Binary protected-label noise was simulated by flipping a fixed fraction of sex labels. The rates were 0.00, 0.05, 0.10, 0.20, 0.30, 0.40, and 0.50. Three noise settings were evaluated: calibration noise only, deployment noise only, and calibration-and-deployment noise. In calibration noise only, Threshold Optimizer is fitted with corrupted calibration protected labels but receives clean protected labels at test-time prediction. In deployment noise only, Threshold Optimizer is fitted with clean calibration labels but receives corrupted protected labels during test-time prediction. The combined setting corrupts both stages.

Missingness was simulated by replacing protected labels with Unknown at rates 0.00, 0.10, 0.20, and 0.30 in the calibration-and-deployment missingness setting. Final fairness evaluation always used the clean protected attribute. This rule is central to the research question: the study measures whether the mitigation remained fair for the real groups, not merely whether it appeared fair under corrupted labels.

Experimental Protocol

Each run used a stratified 60/20/20 train, calibration, and test split. The random seeds were 0, 1, 2, 3, and 4. For each seed and model, the base classifier was fitted on the training split, evaluated unmitigated on the test split, then passed to Threshold Optimizer. Threshold optimization was fitted on the calibration split and evaluated on the final test split. Raw seed-level metrics were saved before summarization, and summary tables report means and standard deviations across the five seeds.

Evaluation Metrics

Predictive metrics were accuracy, balanced accuracy, precision, recall, F1 score, and selection rate. Demographic parity difference was computed as the difference between group selection rates. Equalized odds difference was computed from group differences in true positive and false positive rates. Subgroup outputs

included true positive rate, false positive rate, false negative rate, subgroup accuracy, and confusion-matrix counts. All fairness metrics in the manuscript use clean protected labels for evaluation.

Reproducibility

The executed environment recorded Python 3.12.13, scikit-learn 1.8.0, Fairlearn 0.13.0, NumPy 2.4.4, pandas 3.0.2, matplotlib 3.10.9, and platform macOS-15.5-arm64-arm-64bit. The pipeline saved results/metrics_raw.csv, results/metrics_summary.csv, results/subgroup_metrics.csv, results/experiment_config.json, and the figure files used in this manuscript. The completed raw file contains 510 experiment rows and all rows have completed status. All numeric claims in the manuscript are traceable to those saved outputs.

IV. RESULTS AND DISCUSSION

Baseline Performance

Table I reports the unmitigated classifier results. Random Forest had higher mean accuracy than Logistic Regression in this experiment, while both unmitigated models showed non-trivial demographic parity and equalized odds differences. These baselines are important because post-processing should be interpreted relative to the score distributions and errors produced by the underlying model.

Table I: Baseline unmitigated model results.

Model	*Accuracy*	*Balanced acc.*	*F1*	*DP diff.*	*EO diff.*
LR	0.851 ± 0.002	0.763 ± 0.003	0.656 ± 0.004	0.175 ± 0.008	0.095 ± 0.020
RF	0.862 ± 0.002	0.768 ± 0.006	0.670 ± 0.008	0.171 ± 0.007	0.105 ± 0.030

Note: LR = Logistic Regression; RF = Random Forest; DP diff. = demographic parity difference; EO diff. = equalized odds difference. Values are mean ± standard deviation across five seeds.

Effect of Protected-Attribute Noise

Table II compares clean protected labels with the strongest protected-label flipping condition in which both calibration and deployment labels were corrupted at rate 0.50. Under clean labels, demographic-parity thresholding reduced demographic parity difference sharply for both models, but it increased equalized odds difference. This is expected because the demographic parity constraint controls selection rates rather than conditional error rates. Conversely, clean equalized-odds thresholding reduced equalized odds difference but did not minimize demographic parity difference.

Table II: Selected primary full-builder results (average microseconds per query).

Model	*Constraint*	*Protected labels*	*Accuracy*	*DP diff.*	*EO diff.*
LR	DP	Clean 0.00	0.830 ± 0.002	0.008 ± 0.004	0.348 ± 0.009
LR	DP	High noise 0.50	0.850 ± 0.003	0.187 ± 0.014	0.101 ± 0.022
LR	EO	Clean 0.00	0.834 ± 0.002	0.093 ± 0.004	0.009 ± 0.005
LR	EO	High noise 0.50	0.846 ± 0.005	0.175 ± 0.008	0.091 ± 0.017
RF	DP	Clean 0.00	0.841 ± 0.001	0.007 ± 0.005	0.330 ± 0.013
RF	DP	High noise 0.50	0.862 ± 0.001	0.187 ± 0.014	0.113 ± 0.032
RF	EO	Clean 0.00	0.849 ± 0.002	0.098 ± 0.006	0.017 ± 0.014
RF	EO	High noise 0.50	0.861 ± 0.002	0.191 ± 0.010	0.120 ± 0.026

Note: Constraint abbreviations are DP = demographic parity and EO = equalized odds. High noise refers to 0.50 label flipping during both calibration and deployment. Values are mean ± standard deviation across five seeds.

At high protected-label noise, fairness mitigation became less effective when evaluated against the clean protected attribute. Accuracy sometimes appeared to improve under heavy noise because the mitigation became weaker and moved closer to the unmitigated classifier. This pattern should not be interpreted as fairness improvement; it indicates that the fairness constraint was no longer being reliably applied to the intended groups.

Figure 1 shows the accuracy trend under calibration-and-deployment noise, while Figures 2 and 3 show the corresponding fairness-metric trends. Calibration noise can learn thresholds for wrong groups, deployment noise can apply otherwise valid thresholds to wrong individuals or groups, and the combined setting includes both failure modes.

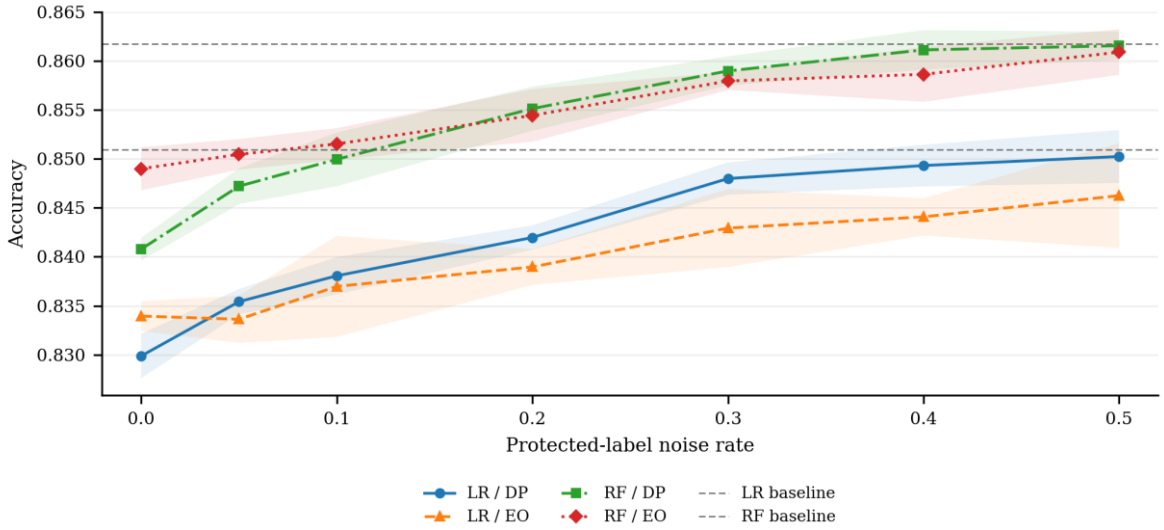

Figure 1: Accuracy versus protected-attribute noise rate.

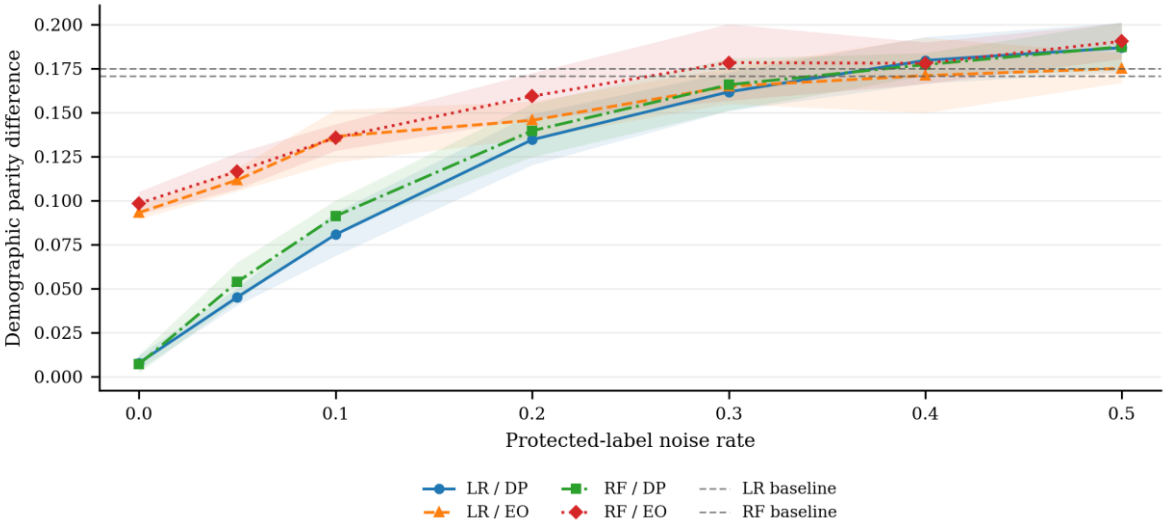

Figure 2: Demographic parity difference versus protected-attribute noise rate.

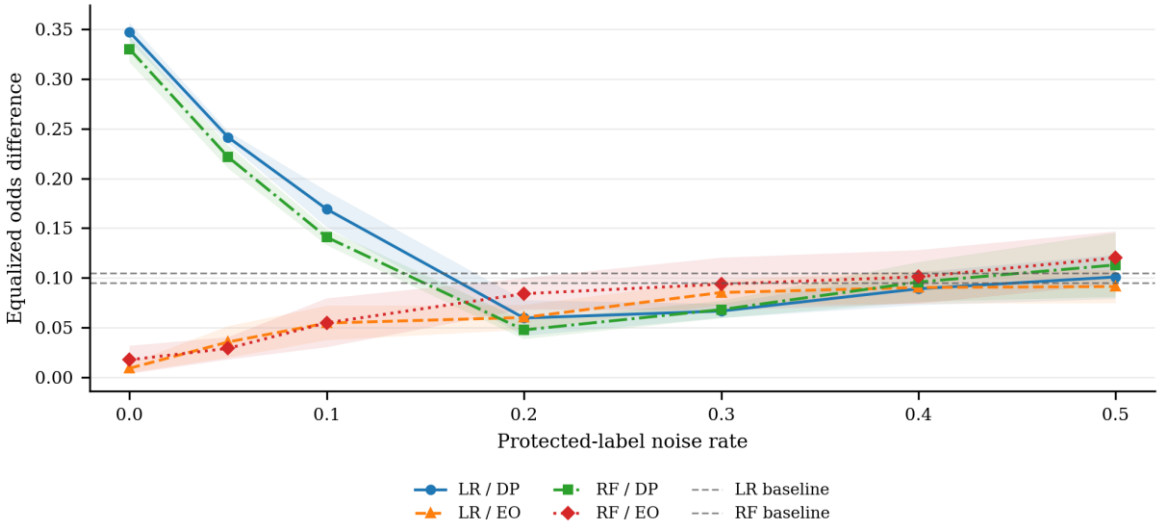

Figure 3: Equalized odds difference versus protected-attribute noise rate.

Effect of Missing Protected Attributes

Table III summarizes missingness experiments. Representing unavailable protected labels as Unknown stabilized the execution of Threshold Optimizer, but it did not guarantee fairness for the clean groups. This distinction matters because an Unknown category can make a system operationally convenient while still failing to preserve fairness for the real protected groups.

Table III**: Threshold Optimizer under protected-attribute missingness.**

Model	*Constraint*	*Missingness*	*Accuracy*	*DP diff.*	*EO diff.*
LR	DP	No missing 0.00	0.830 ± 0.002	0.008 ± 0.005	0.347 ± 0.011
LR	DP	Missing 0.30	0.837 ± 0.001	0.043 ± 0.007	0.232 ± 0.015
LR	EO	No missing 0.00	0.832 ± 0.001	0.089 ± 0.007	0.024 ± 0.024
LR	EO	Missing 0.30	0.835 ± 0.004	0.112 ± 0.008	0.032 ± 0.016
RF	DP	No missing 0.00	0.841 ± 0.001	0.006 ± 0.004	0.332 ± 0.012
RF	DP	Missing 0.30	0.848 ± 0.001	0.054 ± 0.008	0.184 ± 0.012
RF	EO	No missing 0.00	0.849 ± 0.003	0.100 ± 0.009	0.024 ± 0.011
RF	EO	Missing 0.30	0.850 ± 0.004	0.116 ± 0.005	0.042 ± 0.028

Note: Missingness was represented as Unknown during mitigation. Final fairness metrics were evaluated with clean sex labels. Values are mean ± standard deviation across five seeds.

Demographic Parity versus Equalized Odds

The results illustrate the usual constraint-specific trade-off. Demographic parity controls selection rates and can therefore reduce demographic parity difference even when conditional error rates remain unequal. Equalized odds controls true positive and false positive behavior and can therefore reduce equalized odds difference while leaving selection-rate disparities. The study does not rank one definition as universally superior; rather, it shows that protected-label corruption can weaken either constraint when the method depends on protected attributes.

Model-Level Differences

Threshold Optimizer operates on score distributions from the base estimator. Logistic Regression and Random Forest therefore provide different inputs to the post-processor. The final behavior is a joint outcome of preprocessing, base model, calibration split, fairness constraint, protected-label corruption, and test-time sensitive-feature availability. Figure 4 shows subgroup accuracy under demographic-parity post-processing at clean and high-noise settings. The subgroup gap is not fully summarized by overall accuracy, which reinforces the need to inspect group-level diagnostics.

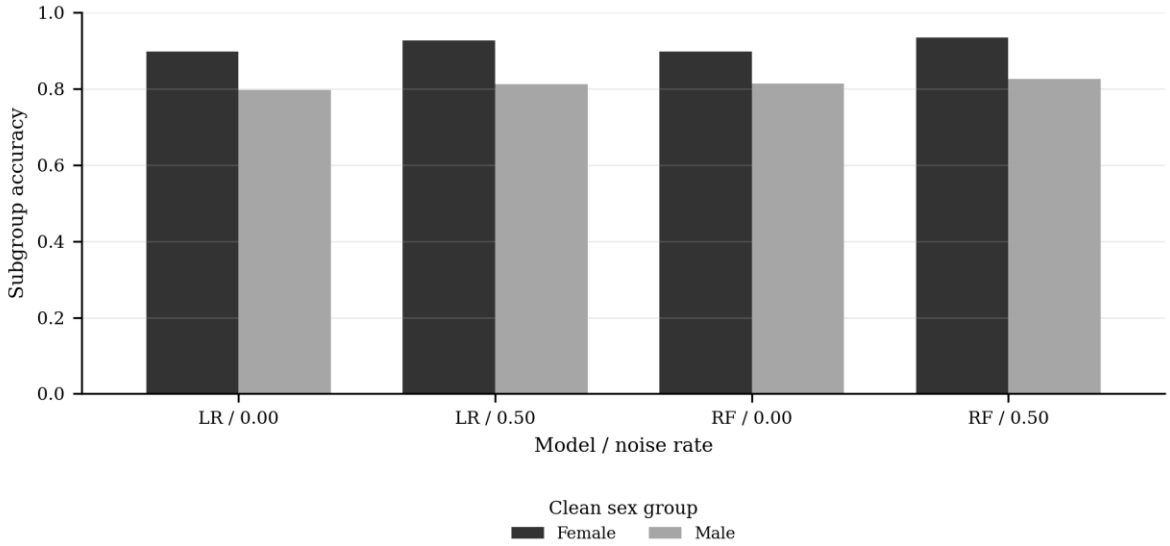

Figure 4: Subgroup accuracy for demographic-parity post-processing at clean and high-noise settings.

Practical Engineering Implications

The engineering implication is direct: protected-attribute quality is part of the fairness-control system. Teams that deploy post-processing mitigation should audit how protected attributes are collected, linked, transformed, stored, and made available at prediction time. Fairness monitoring should use clean or validated protected labels when possible, and uncertainty or missingness should be reported rather than hidden inside an operational category.

Limitations and threats to validity

This study uses one dataset, one primary protected attribute, two base classifiers, and one mitigation method. The Adult dataset is public and reproducible, but it is also an older benchmark built from historical census data. Its labels and protected attributes reflect social and measurement processes from that context and should not be treated as a direct proxy for modern income-decision systems.

The protected-attribute corruption is simulated rather than observed in a deployed system. The binary flipping design is useful for controlled stress testing, but real errors may be systematic, subgroup-dependent, or correlated with other features. The missingness experiment treats missing values as Unknown, which is only one operational strategy. The study also focuses on binary sex as recorded in the dataset and does not address intersectional protected groups.

Although sex and race were removed from the model input features, proxy variables remain. Relationship, marital status, occupation, education, and native country may encode social patterns correlated with protected groups. The findings may not generalize to other domains, other fairness definitions, or mitigation methods that do not require protected attributes at prediction time.

Finally, the executed code pins data splits and protected-attribute corruption through random seeds and saved outputs. Future public release should also expose a repository URL and may set a Threshold Optimizer prediction random_state where supported by the Fairlearn version.

V. CONCLUSION

This study evaluated the robustness of threshold-based fairness post-processing when protected attributes are corrupted or missing. In the executed Adult/Census Income pipeline, clean protected labels allowed Fairlearn ThresholdOptimizer to reduce the fairness gap targeted by the selected constraint. Demographic-parity post-processing reduced the demographic parity difference for both Logistic Regression and Random Forest, while equalized-odds post-processing reduced the equalized odds difference. These results support the central finding that the post-processor can be useful when the protected attribute is available and correctly recorded.

The stress test also shows why protected-label quality is a mathematical reliability condition, not only an administrative data-quality issue. Let A denote the clean protected attribute and A* denote the corrupted attribute used by the post-processor. Threshold Optimizer estimates group-conditioned decision rules using moments defined over A*. Under binary label flipping with rate eta, the observed calibration group A* is a mixture of the true groups: records with A* = a contain approximately (1 - eta) of the true group a and eta of the other group. As eta approaches 0.50, A* becomes increasingly uninformative about A. The learned thresholds therefore satisfy constraints on mixed groups rather than on the clean groups used for evaluation.

A second failure mode appears at deployment. If the learned rule is indexed by A* rather than A, an individual from one true group can receive the threshold intended for the other group. In expectation, the deployed policy becomes a mixture of the intended group-specific thresholds. This weakens the post-processing mechanism and explains why accuracy can appear to improve under extreme protected-label noise: the corrupted mitigation behaves closer to the unmitigated classifier. That pattern is not a fairness improvement when fairness is evaluated with clean protected labels.

Missingness creates a related problem. Treating missing protected labels as Unknown allows the algorithm to run, but the Unknown group is not a social group with stable fairness meaning. It may be a mixture of Female and Male records, and the mixture can change across calibration and deployment. A constraint learned over Female, Male, and Unknown labels therefore does not imply the same constraint over the clean Female and Male groups.

For data engineers, the practical recommendation is to treat protected-attribute handling as part of the fairness-control system. Engineering teams should document protected-attribute lineage, measure missingness and suspected misclassification, test calibration-noise and deployment-noise scenarios separately, and monitor fairness using validated protected labels whenever possible. They should also define fallback behavior for unavailable protected attributes, maintain reproducible seed-level stress tests, audit proxy variables after removing direct protected features, and set minimum data-quality thresholds before relying on post-processing mitigation in production workflows.

VI. DATA AND CODE AVAILABILITY STATEMENT

The Adult/Census Income dataset used in this study is publicly available from the UCI Machine Learning Repository. Processed data, experimental outputs, and the code used for preprocessing, training, bias mitigation, and evaluation are available in the project files accompanying this manuscript. The public repository containing the implementation and related materials is available at: https://github.com/svanidzen-gtu/noisy-protected-attributes-ajer-artifact

REFERENCES

[1]. C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, pp. 214-226, 2012, doi: 10.1145/2090236.2090255.

[2]. M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," Advances in Neural Information Processing Systems 29, pp. 3315-3323, 2016.

[3]. G. Pleiss, M. Raghavan, F. Wu, J. Kleinberg, and K. Q. Weinberger, "On fairness and calibration," Advances in Neural Information Processing Systems 30, 2017.

[4]. A. Agarwal, A. Beygelzimer, M. Dudik, J. Langford, and H. Wallach, "A reductions approach to fair classification," Proceedings of the 35th International Conference on Machine Learning, PMLR 80, pp. 60-69, 2018.

[5]. L. E. Celis, L. Huang, V. Keswani, and N. K. Vishnoi, "Fair classification with noisy protected attributes: A framework with provable guarantees," Proceedings of the 38th International Conference on Machine Learning, PMLR 139, pp. 1349-1361, 2021.

[6]. A. Ghosh, P. Kvitca, and C. Wilson, "When fair classification meets noisy protected attributes," Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society, pp. 679-690, 2023, doi: 10.1145/3600211.3604707.

[7]. A. Shah, M. Shen, J. Ryu, S. Das, P. Sattigeri, Y. Bu, and G. W. Wornell, "Group fairness with uncertain sensitive attributes," IEEE International Symposium on Information Theory, pp. 208-213, 2024, doi: 10.1109/ISIT57864.2024.10619400.

[8]. S. Barocas, M. Hardt, and A. Narayanan, Fairness and Machine Learning: Limitations and Opportunities. Cambridge, MA: MIT Press, 2023.

[9]. S. Bird, M. Dudik, R. Edgar, B. Horn, R. Lutz, V. Milan, M. Sameki, H. Wallach, and K. Walker, "Fairlearn: A toolkit for assessing and improving fairness in AI," Microsoft Technical Report MSR-TR-2020-32, 2020.
[10]. R. K. E. Bellamy et al., "AI Fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias," IBM Journal of Research and Development, vol. 63, no. 4/5, pp. 4:1-4:15, 2019, doi: 10.1147/JRD.2019.2942287.
[11]. B. Becker and R. Kohavi, "Adult," UCI Machine Learning Repository, 1996, doi: 10.24432/C5XW20.
[12]. F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.