HAMLET4Fairness: Enhancing Fairness in AI Pipelines Through Human-Centered AutoML and Argumentation

Joseph Giovanelli[1], **Giuseppe Pisano**[1], **Roberta Calegari**[1]

[1]Alma Mater Studiorum – University of Bologna
j.giovanelli@unibo.it, g.pisano@unibo.it, roberta.calegari@unibo.it

Abstract

AI systems can perpetuate and amplify existing biases and discrimination, prompting academic efforts to develop mitigation techniques. Despite progress, real-world deployments often expose limitations in current methods and tools— overlooking preprocessing, adopting poor evaluation protocols, and failing to integrate domain knowledge. These gaps hinder the effectiveness and reproducibility of fairness solutions. AutoML has emerged as a promising approach to optimize AI pipelines and provide an evaluation framework. However, challenges persist, especially around: intersectionality support, explainability, and stakeholder engagement, which are crucial for fairness and human-centric AI development. We introduce HAMLET4Fairness, integrating AutoML with human-centered approaches grounded in logic and argumentation. This enhances interactivity and transparency in AI pipeline optimization while supporting intersectional fairness. HAMLET4Fairness leverages multi-objective optimization and bounds the search space by user-defined constraints, adapting the CRISP-DM methodology for co-design and collaborative problem solving. We validate HAMLET4Fairness through the well-known case studies in the literature and provide insights into how preprocessing choices affect fairness.

Code — https://github.com/aaai-26/HAMLET4Fairness

Introduction

Artificial intelligence (AI) systems can exacerbate existing societal inequalities and discriminatory patterns (Zuiderveen Borgesius et al. 2018), prompting the development of fairness-aware methods across the AI community. Despite substantial research, current solutions often struggle to generalize beyond controlled settings (Mehrabi et al. 2022; Caton and Haas 2024), due to unrealistic assumptions – such as clean, balanced datasets – and a lack of methodological rigor in model development. Many works do not adequately integrate preprocessing decisions into the fairness evaluation, nor adopt standardized methodologies. Common limitations include: (i) lack of structured frameworks for *human involvement*, preventing co-design of fairness objectives; (ii) over-reliance on single train/test splits instead of robust evaluation protocols such as stratified cross-validation (Weerts et al. 2024); (iii) subjective hyperparameter tuning that undermines reproducibility. Additionally, intersectionality – the impact of multiple sensitive attributes – is often overlooked.

Automated Machine Learning (AutoML) (Karmaker et al. 2021) has emerged as a promising tool, offering efficient pipeline exploration and rigorous evaluation strategies. However, existing AutoML systems are rarely designed with fairness or human-centric principles. They lack mechanisms to incorporate *domain knowledge*, explain decisions, or engage diverse stakeholders, leading to black-box pipelines that optimize accuracy over fairness (Weerts et al. 2024).

This work introduces **HAMLET4Fairness**, an extension of the HAMLET framework (Human-centered AutoML via Logic and Argumentation) (Francia, Giovanelli, and Pisano 2023). While HAMLET combines AutoML with symbolic logic and argumentation for explainable, human-in-the-loop learning, it is not equipped for *multi-objective optimization* nor explicit co-creation practices for fair and trustworthy AI. HAMLET4Fairness brings two main innovations: ***i)*** **Fairness-aware AutoML with intersectionality**—HAMLET4Fairness makes fairness a first-class optimization objective in the AutoML pipeline, supporting constrained multi-objective optimization, balancing performance and fairness metrics, improving robustness with best practices (e.g., stratified k-fold cross-validation) (Weerts et al. 2024), and integrating intersectional fairness evaluation to address discrimination affecting subgroups with multiple sensitive attributes (e.g., gender *and* race); ***ii)*** **Co-creation, user-interaction and transparency**—HAMLET4Fairness promotes a paradigm of interactive, explainable AutoML through a structured argumentation layer. Users can inject, validate, and revise knowledge and constraints, enabling transparent decision-making and facilitating the *co-design* and *co-creation* (Bondi et al. 2021) of fairness strategies. This is implemented through an extension of the CRISP-DM model, operationalized via a symbolic knowledge base and an argumentation graph that evolves across iterations.

The framework is validated through well-known case studies in the literature, showing that HAMLET4Fairness improves the fairness outcomes and predictive performance while producing a transparent representation of the optimization underneath. We also derive actionable insights on

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

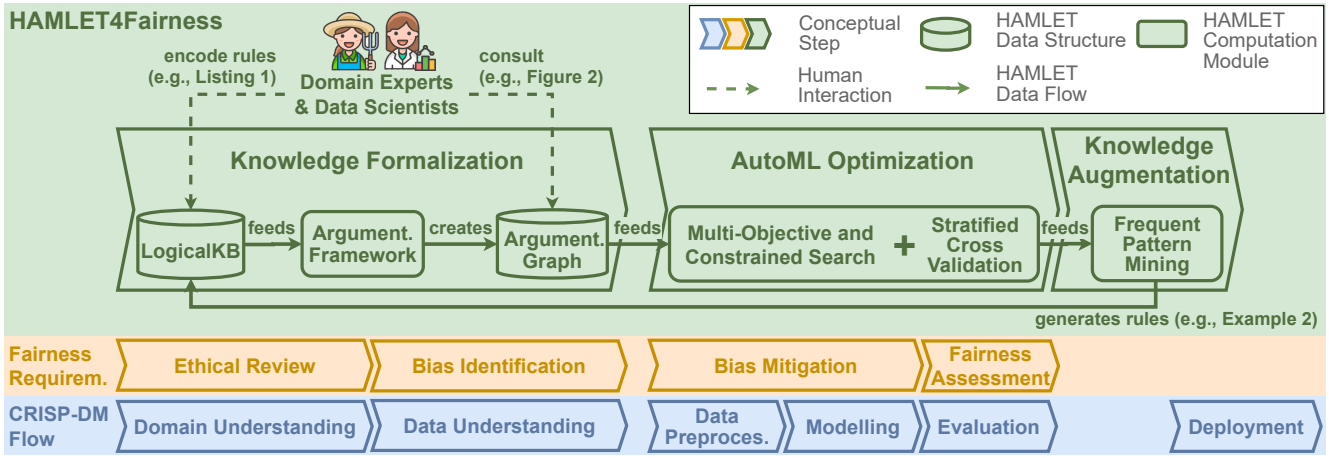


Figure 1: HAMLET4Fairness integrated with the CRISP-DM model and fairness requirements.

how preprocessing steps and mitigation strategies interact with fairness, guiding responsible AI pipeline design. A visual integration with CRISP-DM is shown in Figure 1.

Background

In this section, we introduce the three core dimensions of our work: AutoML, Fairness in AI, and Structured Argumentation. We focus on the supervised task of learning a function $f : \mathbb{X} \to \mathbb{Y}$ from a dataset $\mathcal{D} = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N} \in \mathbb{D} \subset \mathbb{X} \times \mathbb{Y}$, where $\mathbb{X}$ is the instance space, and $\mathbb{Y}$ denotes the output to predict.

Automated Machine Learning

A learning **algorithm** A is fed with instances from $\mathcal{D}$, performs training to tune parameters, and provides a model $H \in \mathbb{H}$ for the task. Algorithms expose hyperparameters controlling the learning process, with the space defined as $\boldsymbol{\Lambda}_A = \Lambda_1 \times \cdots \times \Lambda_M$. Data require preprocessing, organized into a pipeline of steps, transforming D into D'. Each **pipeline** P can differ in order and type of steps. In each **step** S, a transformation T is selected and its hyperparameters tuned. The pipeline's hyperparameter space is $\boldsymbol{\Lambda}_P = \boldsymbol{\Lambda}_{S_1} \times \cdots \times \boldsymbol{\Lambda}_{S_{|P|}}$.

Considering multiple pipelines $\mathcal{P}$ and algorithms $\mathcal{A}$, the search space is $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}_{\mathcal{P}} \times \boldsymbol{\Lambda}_{\mathcal{A}}$. We evaluate a configuration $\lambda \in \Lambda$ with a quality metric $\mathcal{M} : \mathbb{H} \times \mathbb{D} \to \mathbb{R}$, quantifying how well $H = \langle \mathcal{P}, \mathcal{A} \rangle_{\boldsymbol{\lambda}}$ performs on $\mathcal{D}$. The **hyperparameter optimization** (HPO) problem is:

$$\boldsymbol{\lambda}^* \in \arg\max_{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}} \frac{1}{k} \sum_{i=1}^{k} \mathcal{M}\left(\langle \mathcal{P}, \mathcal{A} \rangle_{\boldsymbol{\lambda}}(\mathcal{D}_{train}^{(i)}), \mathcal{D}_{valid}^{(i)}\right)$$

where $\mathcal{D}_{train}^{(i)}$ and $\mathcal{D}_{valid}^{(i)}$ are splits for k-fold cross-validation. State-of-the-art HPO techniques use Bayesian optimization or evolutionary approaches; Bayesian optimization constructs a surrogate model of the objective function and iteratively selects promising configurations.

Fairness in AI

AI models trained on real-world datasets $\mathcal{D}$ can replicate and amplify biases. If gender X_s is a sensitive feature, a model may learn biased patterns, leading to discriminatory outcomes. These risks require fairness-aware design. Fairness interventions can be categorized as: *(i)* preprocessing (modify $\mathcal{D}$ before training), *(ii)* in-processing (adapt A during training), *(iii)* post-processing (alter predictions after training). Fairness is commonly evaluated using metrics comparing model behavior across subpopulations defined by X_s. Widely adopted are: **Demographic Parity (DP)** (Weerts et al. 2023): probability of positive outcome is equal across groups; and **Equalized Odds (EO)** (Weerts et al. 2023): true positive and false positive rates consistent across groups. We use associated metrics like DP ratio and EO ratio (Weerts et al. 2023) for disparities between the most and least advantaged groups. Fairness metrics and interventions are often applied one dimension at a time; this may fail to capture compounded discrimination at the intersection of multiple attributes. An **intersectional perspective** (as in (Crenshaw 1989)) considers individuals belonging to multiple marginalized groups (e.g., women of color) who may experience unique and amplified biases.

One could solve the HPO problem using a **constrained hyperparameter space** $c(\Lambda)$, filtering for fairness. However, this does not actively optimize fairness as its own objective. Balancing fairness with performance leads to a **multi-objective optimization** problem, seeking a Pareto front where solutions are non-dominated. Multi-objective optimization often uses scalarisation methods (e.g., weighted sums). ParEGO (Knowles 2006) applies Bayesian optimization, varying the importance of the multiple objectives to explore diverse parts of the space. Pareto front quality is measured by indicators like Hypervolume (Zitzler and Thiele 1999).

Structured Argumentation

Argumentation addresses the lack of meaningful feedback in current AutoML tools by presenting expert knowledge

and outcomes in a human-readable format. Argumentation guides optimization with domain knowledge and manages the inconsistency common in evolving ML processes. Knowledge from the search space, co-creation, and fairness constraints is encoded as a knowledge graph using an argumentation language. This graph identifies ML pipelines and conflicts, facilitating valid argument identification for AutoML optimization. Structured argumentation provides formal tools to translate human knowledge into arguments.

An **argumentation theory**, AT $= \langle L, R, \sigma, \succ \rangle$, consists of language L, defeasible rules R, conflict function σ, and ordering $\succ$. Each rule $r : \phi_0, \ldots, \phi_n \Rightarrow \phi$ has premises $\phi_0, \ldots, \phi_n \in L$ and conclusion ϕ. The conflict function σ maps formulas to conflicting ones. **Arguments** are constructed recursively. Given arguments $Ar_1, \ldots, Ar_n$ with conclusions $\phi_0, \ldots, \phi_n$, a new argument $Ar = Ar_1, \ldots, Ar_n \Rightarrow \psi$ is constructed if $r : \phi_0, \ldots, \phi_n \Rightarrow \psi \in R$. The conclusion of Ar is $\text{Conc}(Ar) = \psi$, its top rule is $\text{TopRule}(Ar) = r$ and supporting arguments are $\text{Sub}(Ar) = \text{Sub}(Ar_1) \cup \ldots \cup \text{Sub}(Ar_n) \cup \{Ar\}$. Arguments **attack** each other based on σ and $\succ$. Ar_1 attacks Ar_2 if $\text{Conc}(Ar_2') \in \sigma(\text{Conc}(Ar_1))$ for some $Ar_2' \in \text{Sub}(Ar_2)$, provided Ar_2' is not preferred to Ar_1. Preference is determined by the rule ordering. An **Argumentation Graph** for AT is a directed graph $(\mathcal{V}, \rightsquigarrow)$ with vertices $\mathcal{V}$ as arguments and edges defined by attack. $Ar_1 \in \mathcal{V}$ is *acceptable* with respect to $S \subseteq \mathcal{V}$ if, for every argument Ar_2 attacking Ar_1, there exists $Ar_3 \in S$ attacking Ar_2. S is *conflict-free* if no arguments attack each other, and *admissible* if conflict-free and all its arguments are acceptable w.r.t. S. S is a *complete* extension if it is admissible and contains all arguments acceptable w.r.t. itself. The *grounded* extension is the smallest of the complete extensions.

HAMLET4Fairness

Developing AI systems is an iterative process, guided by the CRISP-DM model (Wirth and Hipp 2000). Each iteration engages data scientists and domain experts in refining the solution as new constraints emerge, related to the domain, data, preprocessing, and AI algorithms. Adapting the CRISP-DM methodology to incorporate fairness into AutoML involves integrating co-design and co-creation phases into the process. Figure 1 shows the essential stages for automating fairness with a human-centered approach, allowing stakeholders to make informed decisions. These stages—ethical review, bias identification, bias mitigation, and fairness assessment—address social, legal, and technical requirements. A collaborative co-design process supports these stages, engaging all relevant stakeholders in shaping the system.

HAMLET4Fairness addresses this need by leveraging an argumentative layer atop the AutoML pipeline. It enables users to interact with the system through an Argumentation Graph built upon a Logical Knowledge Base (LogicalKB). This knowledge base allows for the injection of requirements identified during the CRISP-DM phases and facilitates reciprocal interaction, where the AutoML system can demonstrate what it has learned from various experiments and pipeline comparisons. At the end of the AutoML experimentation, in which different pipelines are applied and evaluated for both fairness and accuracy, the LogicalKB is enriched with insights gained throughout the process. HAMLET4Fairness operates in the following iterative phases.

Knowledge Formalization. This phase is a co-design and co-creation process involving domain experts and data scientists engaged in the AI system's decision-making processes. Participants collaboratively write the LogicalKB. During this phase, based on the envisioned constraints, it is possible to build the foundational Argumentation Graph and check compliance with all applicable constraints, automatically generating the search space for the AutoML tool. Preliminary conflicts can be solved here. If this step proceeds to phase 2 without initializing the LogicalKB, AutoML will explore the entire search space.

AutoML Optimization. During the data preprocessing, modeling, and evaluation stages, AutoML explores different solutions within the constrained search space through multi-objective optimization. This enables the process to seek the best accuracy as a learning objective, while also achieving results in terms of fairness. This phase retrieves the best solutions used in the next step to induce new logical knowledge that augments the LogicalKB.

Knowledge Augmentation. The exploration of different pipelines may reveal new constraints that further narrow the search space (e.g., it is pointless to insert a feature engineering step if you have not performed a data mitigation step first). Data scientists and domain experts evaluate the newly generated knowledge, updating the LogicalKB with additional constraints or any changes deemed appropriate, then starting a new iteration of the CRISP-DM process.

Knowledge Formalization

The HPO problem encodes the AutoML goal: given a set of preprocessing steps and classification algorithms, we want to find a pipeline of steps and the values of their hyperparameters that optimize our objectives (metrics). To formalize pipeline configurations and associated constraints, HAMLET4Fairness defines a logical language. Let L be this AutoML argumentation language that allows users to encode the pipeline base elements: `step(`S`)`, representing a step S in the pipeline; `algorithm(`S, A`)`, representing an algorithm A for the step S; `hyperparameter(`A, h, t`)`, representing a hyperparameter h for the algorithm A of type t; `domain(`A, h, Λ_h`)`, representing the domain Λ_h of the hyperparameter h for the algorithm A, i.e., the range of values that the hyperparameter can assume; `pipeline(`$\langle S_1, \ldots, S_n \rangle, A$`)`, representing a pipeline consisting of the sequence of steps $\langle S_1, \ldots, S_n \rangle$ and the classification algorithm A.

To encode constraints, the framework allows the following predicates: `mandatory(`$\langle S_1, \ldots, S_n \rangle, A$`)`, imposing the steps $\langle S_1, \ldots, S_n \rangle$ on the pipelines with algorithm A; `forbidden(`$\langle S_1, \ldots, S_n \rangle, A$`)`, forbidding any of the steps $\langle S_1, \ldots, S_n \rangle$ in pipelines with A; `mandatory_order(`$\langle S_1, \ldots, S_n \rangle, A$`)`, imposing the order of steps $\langle S_1, \ldots, S_n \rangle$ on pipelines with A. Fairness constraints that forbid certain steps can be encoded, such as normalization not being legally valid in certain contexts.

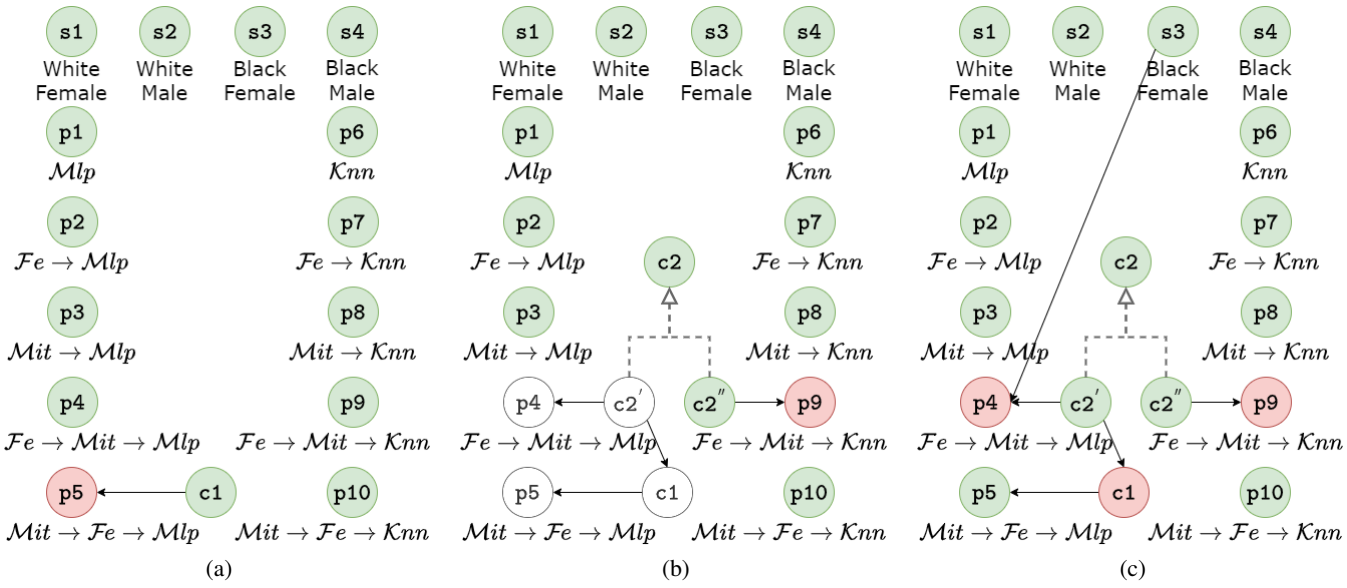


Figure 2: Examples of Argumentation Graphs, including: possible pipelines (`p1`, `p2`, ...), constraints (`c1`, `c2`), and sensitive groups (`s1`, `s2`, ...). Green nodes are valid arguments, red ones are refuted, white ones are frozen. Arrows are attacks.

Conflicts in the AutoML language are managed by a conflict function: a pipeline `pipeline(`$\langle S_i, S_j \rangle$`, A)` conflicts with `mandatory` constraints if required steps are missing, with `forbidden` if containing forbidden steps, and with `mandatory_order` if steps are out of order. Conflicts also arise between `mandatory` and `forbidden` constraints over shared steps, and between different `mandatory_order` listings. The language L supports fairness-aware modelling by representing sensitive attributes via `sensitive_feature(`F`,` $\langle V_1, \ldots, V_n \rangle$`)`, specifying a sensitive feature F and values $V_1, \ldots, V_n$ to be considered in fairness metrics. Sensitive groups—distinct subpopulations from all specified sensitive feature values—are represented as `sensitive_group(`$\langle V_{F_1}, \ldots, V_{F_m} \rangle$`)`. The argumentation language L also supports predicates defining the evaluation context: `dataset(`D`)` sets the dataset D; `metric(`M_P`)` declares the performance metric M_P; `fairness_metric(`M_F`)` encodes the fairness metric M_F. Predicates `performance_thresholds(`θ_x, θ_y`)`, `fairness_thresholds(`δ_-, δ_+`)`, and `mining_support(`τ`)` set parameters for rule mining. Definitions may conflict when `pipeline` is found to discriminate against a group. This is represented by `discriminate(pipeline(`$\langle S_1, \ldots, S_n \rangle, A$`),` $\langle v_{F_1}, \ldots, v_{F_m} \rangle$`)`, indicating that pipeline $\langle S_1, \ldots, S_n \rangle$ with classifier A discriminates against group $\langle v_{F_1}, \ldots, v_{F_m} \rangle$ with respect to the fairness metric. Adding or removing such `discriminate` predicates dynamically adjusts the empirical evidence about fairness violations.

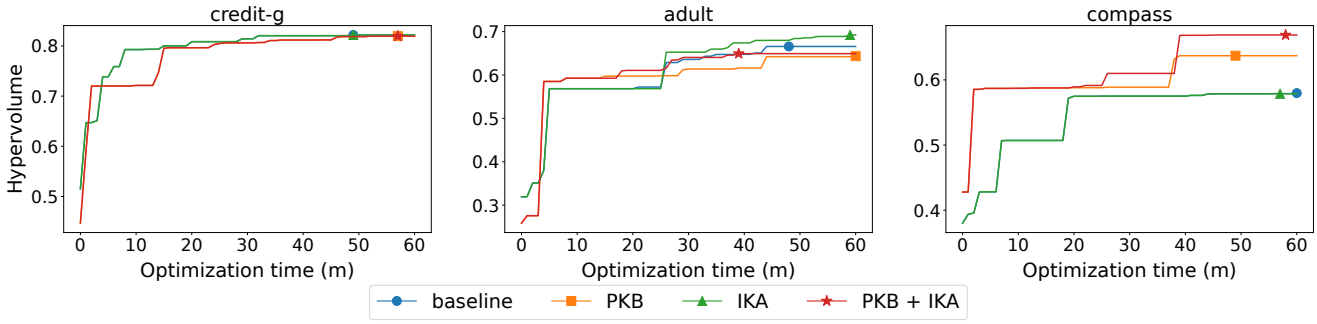
Example 1 (Logical KB). *Let us consider a simple KB. First, we define the problem setting: the dataset for training the pipelines, the performance and fairness metrics, thresholds, and mining support. The pipeline comprises three steps: Feature Engineering ($\mathcal{F}e$), Data Mitigation ($\mathcal{M}it$), and Classification ($\mathcal{C}l$). We only define algorithms: Multilayer Perceptron ($\mathcal{M}lp$) and K-nearest Neighbors ($\mathcal{K}nn$). A fairness constraint `c1` is set, indicating that in pipelines using $\mathcal{M}lp$, $\mathcal{M}it$ should be applied after $\mathcal{F}e$, generating a conflict with pipelines not aligned.*

Listing 1: Example of LogicalKB

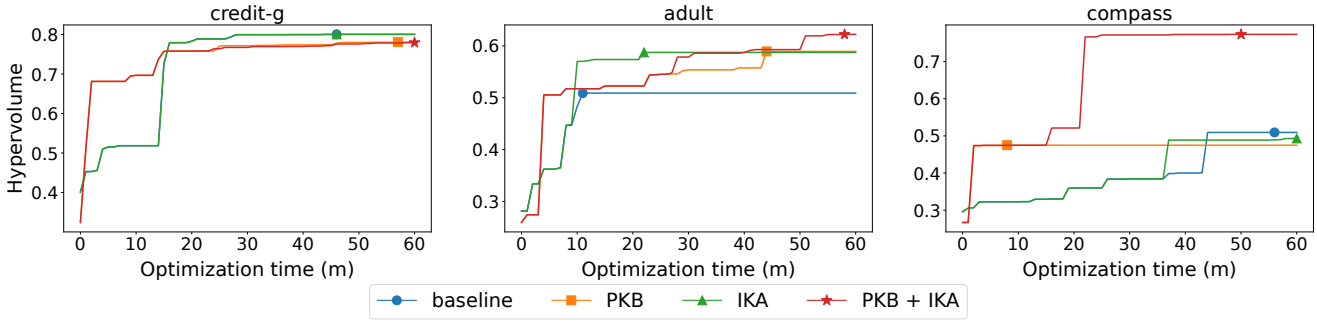
```
% Problem definition
dataset(COMPASS).
performance_metric(balanced_accuracy).
performance_thresholds(0.4, 0.6).
fairness_metric(equalized_odds_ratio).
fairness_tresholds(0.4, 0.6).
mining_support(0.6).
sensitive_feature(Sex, ⟨Female, Male⟩).
sensitive_feature(Race, ⟨Black, White⟩).

% Define steps and algorithms
step(Mit). step(Fe). step(Cl).
algorithm(Cl, Mlp).
algorithm(Cl, Knn).

% Define fairness constraints
c1 : ⇒ mandatory_order(⟨Fe, Mit⟩, Mlp).
```

AutoML optimization

Based on the knowledge base (KB), HAMLET4Fairness constructs an Argumentation Graph to prune regions of the search space represented by `pipeline` predicates. This maps the original optimization space Λ to the constrained subspace $c(\Lambda)$, satisfying all encoded constraints. The graph



(a) Hypervolume over time optimizing Demographic Parity Ratio and Balanced Accuracy.



(b) Hypervolume over time optimizing Equalized Odds Ratio and Balanced Accuracy.

Figure 3: Comparison of fairness optimization strategies under intersectionality constraints.

is evaluated using grounded semantics, i.e., only arguments that the model is certain about are included. With conflicting constraints, neither is included; portions of the space attacked by both can be explored. This approach requires human supervision when conflicting information creates doubt—human experts decide on the next steps.

For Example 1, the initial Argumentation Graph (Figure 2a) explicitly represents feasible pipelines, the Cartesian product of sensitive groups, and applicable constraints with their attacks. Pipeline `p5` is attacked by constraint **c1**, as it violates an ordering constraint.

To explore the constrained hyperparameter space, HAMLET4Fairness adopts ParEGO (Knowles 2006) for multi-objective optimization of performance and fairness. At each iteration, new hyperparameter configurations are proposed to explore diverse and promising regions of the search space. For robust model assessment, stratified k-fold cross-validation is used, stratifying not just by class but also by sensitive attributes (Weerts et al. 2024). This ensures both training and validation folds preserve subgroup distributions. The optimization results in a Pareto front comprising non-dominated solutions, each expressing a different trade-off between performance and fairness.

Knowledge Augmentation

Users may manually inject constraints into the `LogicalKB`. The system also performs rule discovery over explored solutions to identify constraints supporting optimal pipeline construction. Rules capture relationships between steps that frequently co-occur in promising or discouraging pipelines w.r.t. performance and fair metrics of the LogicalKB. This is determined using the performance and fairness thresholds (δ_- and δ_+). For each algorithm (e.g., $\mathcal{M}lp$), frequent itemset mining (Srikant and Agrawal 1997) suggests or discards steps for `mandatory` and `forbidden` constraints; frequent sequence mining (Srikant and Agrawal 1996) derives orderings for `mandatory_order` and `discriminate` constraints. To detect discrimination, we compute fairness metrics by group, not in aggregate, identifying pipelines that exhibit bias toward subpopulations.

Example 2 (Rules Discovery)**.** *From the* `LogicalKB` *in Example 1 and the Argumentation Graph in Figure 2a, AutoML results are analyzed, and rule discovery is applied. With the $\mathcal{K}nn$ algorithm, all pipelines where $\mathcal{F}e$ is applied before $\mathcal{M}it$ fail to meet objectives. Reversing order to $\mathcal{M}it$ before $\mathcal{F}e$ succeeds. A new rule is mined:*

$c2'': \Rightarrow mandatory\_order(\langle \mathcal{M}it, \mathcal{F}e \rangle, \mathcal{K}nn)$.

Humans may generalize the rule:

$c2 : \Rightarrow mandatory\_order(\langle \mathcal{M}it, \mathcal{F}e \rangle, \mathcal{C}l)$.

With this in the KB, HAMLET4Fairness also generates:

$c2': \Rightarrow mandatory\_order(\langle \mathcal{M}it, \mathcal{F}e \rangle, \mathcal{M}lp)$.

This new argument conflicts with `c1` *(Figure 2b). The data scientist may rerun the process for more evidence. If new evidence is mined as:*

$discriminate(pipeline(\langle \mathcal{F}e, \mathcal{M}it \rangle, \mathcal{M}lp), \langle Black, Female \rangle)$.

The `discriminate` *predicate means that pipelines where $\mathcal{F}e$ precedes $\mathcal{M}it$ with $\mathcal{M}lp$ yield poor fairness for group*

$\langle Black, Female \rangle$. *The data scientist may prioritize* c_1 *over* c_2 *using* $\texttt{sup}(c_2, c_1)$*, subordinating* c_2 *if in conflict. The Argumentation Graph is updated accordingly (Figure 2c).*

Validation and Findings

We define three experimental settings for evaluating HAMLET4Fairness: **PKB (Preliminary Knowledge Base)** starts with a preliminary LogicalKB constraining the search space from the first iteration, with no rule mining applied. The LogicalKB includes (i) rules previously discovered in the literature (Giovanelli, Bilalli, and Abelló 2022) about step order for effective pipelines in terms of accuracy (e.g., normalization before feature engineering) and (ii) beliefs from co-design concerning fairness (i.e., a mitigation step should always be present and before other preprocessing). **IKA (Iterative Knowledge Augmentation)** starts with an empty LogicalKB, adding all rules recommended after each run to the KB. **PKB-IKA** starts with a preliminary LogicalKB and adds recommended rules after each run.

We compare these settings against a baseline with no prior constraints or knowledge augmentation. Each approach, including the baseline, has 60 minutes for execution. For knowledge augmentation approaches, this time is divided into four iterations of 15 minutes each. For HAMLET4Fairness, the augmentation mechanism is supervised; for automation, we accept rules whose metric thresholds and support exceed 0.6 as defined in the knowledge base. The search space consists of six pipeline steps: four standard data preprocessing operations—Normalization, Discretization, Feature Engineering, and Rebalancing—plus mitigation and classification. The exploration procedure is implemented using SMAC (Hutter, Hoos, and Leyton-Brown 2011; Lindauer et al. 2022). We evaluate on three benchmark datasets: Credit-g (Hofmann 1994), Adult (Becker and Kohavi 1996), and COMPAS (Dressel and Farid 2018), considering gender and race as sensitive attributes. Multi-objective optimization targets performance—measured by balanced accuracy (Pedregosa et al. 2011)—and fairness, assessed via demographic parity ratio and equalized odds ratio (Weerts et al. 2023). Details on search space, datasets, and metric definitions are in the supplementary materials.

Convergence and Scalability. During optimization, model configurations are evaluated on balanced accuracy and a fairness metric, using the Pareto front of non-dominated models for assessment. We use the hypervolume quality indicator (Zitzler and Thiele 1999) to measure the objective space volume covered by the Pareto front. A higher hypervolume indicates better front spread and convergence. Figure 3b and Figure 3a show hypervolume evolution over time.

Effectiveness. For both demographic parity and equalized odds, at least one HAMLET4Fairness configuration matches or outperforms the baseline, demonstrating improvement in fairness without compromising predictive performance. On *Credit-g*, all four approaches achieve similar performance, but in later optimization, IKA generates rules that meaningfully explain search space exploration. For *Adult* and *COMPAS*, the synergy between PKB and iterative mining emerges: PKB-IKA consistently achieves the highest scores, validating the combined value of pre-encoded and discovered knowledge. Notably, integrating both rule types enables PKB-IKA to confirm preliminary hypotheses, further analyzed in the next section.

Efficiency. IKA and the baseline show similar early trends since both start from an empty `LogicalKB`. IKA improves towards the end via iterative knowledge discovery. In contrast, PKB and PKB-IKA benefit from pre-encoded rules, leading to greater gains—especially in the first 10 minutes. This illustrates the value of domain knowledge in guiding the search early on.

Rule Discovery Throughout the HAMLET4Fairness runs, around 100 constraints were generated according to metric thresholds and support, providing insights into effective regions of the search space. For brevity, we discuss the rules mined with equalized odds in Table 1; the remaining demographic parity results are in the supplementary materials. We group rules by dataset, approach, and constraint type. On Credit-g with IKA, 1. shows the fairness metric suggesting to `forbid` pipelines with $\mathcal{F}e$, $\mathcal{N}$, and $\mathcal{M}it$ for all algorithms. This is confirmed by 2., where the same steps are discarded by $\mathcal{M}lp$ and $\mathcal{R}f$, discriminating Males who are Divorced/Separated. In 3., these steps are suggested with a `mandatory` constraint, highlighting good performance by balanced accuracy. Thus, pipelines without mitigation failed fairness, privileging accuracy. On COMPAS with PKB-IKA, 4. shows all preprocessing steps improve results with $\mathcal{R}f$, especially when $\mathcal{M}it$ and $\mathcal{R}eb$ are applied with $\mathcal{F}e$ or $\mathcal{N}$. According to 5., $\mathcal{M}it$ should always be applied first. These rules validate beliefs in the Preliminary LogicalKB and are effective in guiding optimization towards fair, high-performing pipelines.

Related Works

Data preprocessing Impact on Fairness (Friedler et al. 2019) observed that data preparation significantly affects dataset characteristics and fairness. Building on calls by (Caton and Haas 2024) and (Mehrabi et al. 2022) for more research into fairness impacts of specific techniques such as Principal Component Analysis (PCA), our study investigates feature engineering transformations, namely Select K Best and PCA (Pedregosa et al. 2011), complemented by mitigation methods like Correlation Remover (Weerts et al. 2023) and Learn Fair Representation (Zemel et al. 2013). While (Yang et al. 2020) introduced fair-DAGs to detect biases in preprocessing within a singular pipeline, (Biswas and Rajan 2021) broadened the scope to multiple pipelines. Finally, visualization tools like FairVis (Cabrera et al. 2019) support the detection of subgroup-specific fairness issues. Our research advances beyond these efforts by analyzing varied hyperparameter configurations to enhance fairness assessments during preprocessing and incorporating a human-centered argumentation approach in AutoML.

Fairness in AutoML The integration of fairness into AutoML systems is an emerging focus. Despite this interest, AutoML's limited interactivity and explainability often deter its use. (Wu and Wang 2021) developed FairAutoML,

ID	Dataset	Approach	Constraint	Pipeline			Algorithm	Metric	Group
1	Credit-g	IKA	`forbidden`	$\langle \mathcal{F}e, \mathcal{N}, \mathcal{R}eb \rangle$			$\mathcal{M}lp, \mathcal{R}f, \mathcal{K}nn$	EOR	-
2			`discriminate`	$\langle \mathcal{F}e \rangle$,	$\langle \mathcal{N} \rangle$,	$\langle \mathcal{R}eb \rangle$	$\mathcal{M}lp, \mathcal{R}f$	EOR	Male, Div/Sep
3			`mandatory`	$\langle \mathcal{F}e, \mathcal{N} \rangle$,	$\langle \mathcal{F}e, \mathcal{R}eb \rangle$,	$\langle \mathcal{N}, \mathcal{R}eb \rangle$	$\mathcal{M}lp$	BA	-
4	COMPAS	PKB-IKA	`mandatory`	$\langle \mathcal{M}it, \mathcal{F}e, \mathcal{R}eb \rangle$,	$\langle \mathcal{M}it, \mathcal{N}, \mathcal{R}eb \rangle$		$\mathcal{R}f$	EOR	-
5			`mandatory_order`	$\langle \mathcal{M}it, \mathcal{F}e \rangle$,	$\langle \mathcal{M}it, \mathcal{N} \rangle$,	$\langle \mathcal{M}it, \mathcal{R}eb \rangle$	$\mathcal{R}f$	EOR	-

EOR = Equalized Odds Ratio, BA = Balanced Accuracy, $\mathcal{M}it$ = Mitigation, $\mathcal{N}$ = Normalization, $\mathcal{F}e$ = Feature Eng., $\mathcal{R}eb$ = Rebalancing, $\mathcal{K}nn$ = KNeighborsClassifier, $\mathcal{R}f$ = RandomForestClassifier, $\mathcal{M}lp$ = MLPClassifier

Table 1: Discovered rules for Equalized Odds Ratio.

Topic	HAMLET	HAMLET4FAIRNESS
Focus	Structural pipeline configuration	Fairness-aware configuration, evaluation, and constraint handling
Core Predicates	Steps, algorithms, hyperparameters, domains, structural constraints	HAMLET predicates and fairness-specific constructs
Fairness Modelling	Not supported	Explicit modelling of sensitive features, subgroups, and fairness metrics
Discrimination Representation	Not supported	`discriminate` predicate captures evidence of group-specific unfairness
Evaluation Context	Implicit or external	Explicit (dataset, metric, fairness metric, thresholds)
Conflict Function	Based on structural constraint violations	Extended to include group-level discrimination conflicts
Rule Mining	Based on performance; uses frequent itemset and sequence mining to extract constraints	Same mining techniques extended with support filtering and by-group discrimination detection
Support Thresholds	Not parameterized	User-defined thresholds for performance, fairness, and rule generalizability
Fairness Evaluation Granularity	Not supported	Disaggregated by sensitive group; detects group-specific unfairness patterns

Table 2: Comparison between HAMLET and HAMLET4FAIRNESS

a framework that integrates in-processing bias mitigation strategies within AutoML. (Schmucker et al. 2020) and (Cruz et al. 2021) advance fairness by incorporating it as an objective in multi-objective HPO to balance fairness and accuracy, with the latter allowing users to specify performance-fairness trade-offs. (Perrone et al. 2021) introduced Fair Bayesian Optimization (FairBO), treating fairness as a constraint in a constrained optimization problem. While existing approaches focus on integrating fairness through in-processing techniques during the HPO process, our method enhances complete ML pipelines with preprocessing fairness interventions. We merge multi-objective and constrained optimization for greater effectiveness, guided by (Weerts et al. 2024)'s advocacy for structured processes like CRISP-DM to enhance AutoML fairness. Drawing inspiration from CRISP-DM and employing argumentation, we boost AutoML's interactivity and explainability, thus improving human-centric approaches.

On the differences between HAMLET and HAMLET4Fairness The standard HAMLET implementation leverages a single-objective optimization on pipelines with no mitigation algorithms, evaluated through standard cross-validation. The HAMLET4Fairness framework extends the original HAMLET language to address fairness concerns in automated machine learning. While HAMLET focuses primarily on the structural and functional configuration of pipelines – using logical rules to model steps, algorithms, and performance-driven constraints – HAMLET4Fairness introduces additional constructs to represent fairness metrics, sensitive features and groups, and discrimination patterns. It enhances the reasoning process with disaggregated fairness evaluation, support-based rule generalization, and the ability to detect and exclude discriminatory pipelines. These differences are summarized in Table 2, which contrasts the language expressiveness, evaluation context, and rule mining capabilities of the two frameworks.

Conclusions

We present **HAMLET4Fairness**, an AutoML extension that integrates a symbolic knowledge base and argumentation layer for fairness-aware, interactive machine learning. Built on the CRISP-DM framework, it enables iterative, user-in-the-loop refinement and promotes **co-design and co-creation** for fairness. A key feature is explicit modeling of **intersectionality**, detecting and mitigating bias in subgroups of sensitive attributes. Through argumentation, users can inject, validate, and revise domain knowledge, improving traceability and interpretability. Empirical results across benchmarks show that HAMLET4Fairness improves fairness metrics, such as demographic parity and equalized odds, while maintaining predictive performance and enhancing model robustness and transparency for trustworthiness. Future work includes systematic ablation studies to quantify how pipeline choices impact fairness and performance.

Ethical Statement

The research conducted herein adheres to ethical standards throughout its entirety. All procedures and methodologies employed in this study have been designed and executed to comply with established ethical guidelines. It is important to note that we exclusively utilised publicly available datasets, already recognised as adherent to legal and ethical standards, ensuring the responsible and lawful acquisition of data.

Acknowledgements

This work was supported by the "AEQUITAS" project funded by the European Union's Horizon Europe research and innovation programme under grant number 101070363; and by the European Research Council (ERC) Project "CompuLaw" (Grant Agreement No 833647) under the European Union's Horizon 2020 research and innovation program.

References

Becker, B.; and Kohavi, R. 1996. Adult. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5XW20.

Biswas, S.; and Rajan, H. 2021. Fair preprocessing: towards understanding compositional fairness of data transformers in machine learning pipeline. In Spinellis, D.; Gousios, G.; Chechik, M.; and Penta, M. D., eds., *ESEC/FSE '21: 29th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Athens, Greece, August 23-28, 2021*, 981–993. ACM.

Bondi, E.; Xu, L.; Acosta-Navas, D.; and Killian, J. A. 2021. Envisioning communities: a participatory approach towards AI for social good. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, 425–436.

Cabrera, Á. A.; Epperson, W.; Hohman, F.; Kahng, M.; Morgenstern, J.; and Chau, D. H. 2019. FairVis: Visual analytics for discovering intersectional bias in machine learning. In *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*, 46–56. IEEE.

Caton, S.; and Haas, C. 2024. Fairness in Machine Learning: A Survey. *ACM Comput. Surv.*, 56(7).

Crenshaw, K. 1989. Demarginalizing the Intersection of Race and Sex: A Black Feminist Critique of Antidiscrimination Doctrine, Feminist Theory and Antiracist Politics. *University of Chicago Legal Forum*, 1989: 139–167.

Cruz, A. F.; Saleiro, P.; Belém, C.; Soares, C.; and Bizarro, P. 2021. Promoting Fairness through Hyperparameter Optimization. In Bailey, J.; Miettinen, P.; Koh, Y. S.; Tao, D.; and Wu, X., eds., *IEEE International Conference on Data Mining, ICDM 2021, Auckland, New Zealand, December 7-10, 2021*, 1036–1041. IEEE.

Dressel, J.; and Farid, H. 2018. The accuracy, fairness, and limits of predicting recidivism. *Science advances*, 4(1): eaao5580.

Francia, M.; Giovanelli, J.; and Pisano, G. 2023. HAMLET: A framework for Human-centered AutoML via Structured Argumentation. *Future Generation Computer Systems*, 142: 182–194.

Friedler, S. A.; Scheidegger, C.; Venkatasubramanian, S.; Choudhary, S.; Hamilton, E. P.; and Roth, D. 2019. A comparative study of fairness-enhancing interventions in machine learning. In danah boyd; and Morgenstern, J. H., eds., *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019, Atlanta, GA, USA, January 29-31, 2019*, 329–338. ACM.

Giovanelli, J.; Bilalli, B.; and Abelló, A. 2022. Data preprocessing pipeline generation for AutoETL. *Inf. Syst.*, 108: 101957.

Hofmann, H. 1994. Statlog (German Credit Data). UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5NC77.

Hutter, F.; Hoos, H. H.; and Leyton-Brown, K. 2011. Sequential Model-Based Optimization for General Algorithm Configuration. In Coello, C. A. C., ed., *Learning and Intelligent Optimization - 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. Selected Papers*, volume 6683 of *Lecture Notes in Computer Science*, 507–523. Springer.

Karmaker, S. K.; Hassan, M. M.; Smith, M. J.; Xu, L.; Zhai, C.; and Veeramachaneni, K. 2021. Automl to date and beyond: Challenges and opportunities. *ACM Computing Surveys (CSUR)*, 54(8): 1–36.

Knowles, J. D. 2006. ParEGO: a hybrid algorithm with online landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1): 50–66.

Lindauer, M.; Eggensperger, K.; Feurer, M.; Biedenkapp, A.; Deng, D.; Benjamins, C.; Ruhkopf, T.; Sass, R.; and Hutter, F. 2022. SMAC3: A Versatile Bayesian Optimization Package for Hyperparameter Optimization. *J. Mach. Learn. Res.*, 23: 54:1–54:9.

Mehrabi, N.; Morstatter, F.; Saxena, N.; Lerman, K.; and Galstyan, A. 2022. A Survey on Bias and Fairness in Machine Learning. *ACM Comput. Surv.*, 54(6): 115:1–115:35.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830.

Perrone, V.; Donini, M.; Zafar, M. B.; Schmucker, R.; Kenthapadi, K.; and Archambeau, C. 2021. Fair Bayesian Optimization. In Fourcade, M.; Kuipers, B.; Lazar, S.; and Mulligan, D. K., eds., *AIES '21: AAAI/ACM Conference on AI, Ethics, and Society, Virtual Event, USA, May 19-21, 2021*, 854–863. ACM.

Schmucker, R.; Donini, M.; Perrone, V.; and Archambeau, C. 2020. Multi-objective multi-fidelity hyperparameter optimization with application to fairness. In *NeurIPS 2020 Workshop on Meta-learning*.

Srikant, R.; and Agrawal, R. 1996. Mining Sequential Patterns: Generalizations and Performance Improvements. In Apers, P. M. G.; Bouzeghoub, M.; and Gardarin, G., eds., *Advances in Database Technology - EDBT'96, 5th International Conference on Extending Database Technology,*

Avignon, France, March 25-29, 1996, Proceedings, volume 1057 of *Lecture Notes in Computer Science*, 3–17. Springer.

Srikant, R.; and Agrawal, R. 1997. Mining generalized association rules. *Future Gener. Comput. Syst.*, 13(2-3): 161–180.

Weerts, H.; Dudík, M.; Edgar, R.; Jalali, A.; Lutz, R.; and Madaio, M. 2023. Fairlearn: Assessing and Improving Fairness of AI Systems.

Weerts, H.; Pfisterer, F.; Feurer, M.; Eggensperger, K.; Bergman, E.; Awad, N.; Vanschoren, J.; Pechenizkiy, M.; Bischl, B.; and Hutter, F. 2024. Can fairness be automated? Guidelines and opportunities for fairness-aware AutoML. *Journal of Artificial Intelligence Research*, 79: 639–677.

Wirth, R.; and Hipp, J. 2000. CRISP-DM: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, volume 1. Springer-Verlag London, UK.

Wu, Q.; and Wang, C. 2021. Fair AutoML. *CoRR*, abs/2111.06495.

Yang, K.; Huang, B.; Stoyanovich, J.; and Schelter, S. 2020. Fairness-Aware Instrumentation of Preprocessing Pipelines for Machine Learning. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics (HILDA 2020), co-located with SIGMOD 2020*, 4. Portland, OR, USA: HILDA. Workshop on Human-In-the-Loop Data Analytics : co-located with SIGMOD 2020 : 19 June 2020, Portland, OR, USA.

Zemel, R. S.; Wu, Y.; Swersky, K.; Pitassi, T.; and Dwork, C. 2013. Learning Fair Representations. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, 325–333. JMLR.org.

Zitzler, E.; and Thiele, L. 1999. Multiobjective Evolutionary Algorithms: a Comparative Case Study and the Strength Pareto Approach. *IEEE TEVC*, 3(4): 257–271.

Zuiderveen Borgesius, F.; et al. 2018. Discrimination, artificial intelligence, and algorithmic decision-making. *Council of Europe, Directorate General of Democracy*, 42.