

ORIGINAL RESEARCH

Open access

Toward Fair AI Development: How Developers Can Address Algorithmic Bias

Marco De Luca^{1*}, Giulia Ferraro², Antonio Russo¹, Elena Rossi³

Abstract

Artificial intelligence (AI) has swiftly emerged as a core technology for securing a competitive advantage. Nonetheless, there are escalating anxieties regarding bias within AI models, given that AI developers are susceptible to introducing bias both inadvertently and intentionally. Utilizing a qualitative methodology, this study explored how AI developers can actively foster the creation of equitable AI models. The primary results indicate that the danger of bias stems predominantly from a deficiency in social and gender diversity within AI development teams, alongside pressure from AI managers to rapidly deliver highly anticipated outcomes. Furthermore, the integrity of AI developers is critical, as they possess the ability to hide biases from management and other relevant AI stakeholders. The testing phase preceding model deployment is also prone to bias because it seldom reflects the diverse social demographics that could be impacted.

This research offers actionable recommendations across four core dimensions: governance, social, technical, and training and development procedures. Responsible organizations must implement deliberate measures to guarantee that their AI developers follow equitable protocols during AI creation; AI developers need to put ethical considerations first and evaluate the societal impacts of their models; collaborative partnerships ought to be formed among AI developers, AI stakeholders, and the communities likely to be affected by these models; and AI developers must emphasize transparency and explainability within their models while ensuring robust bias testing and remediation prior to deployment. Lastly, providing emotional intelligence training to AI developers is recommended to equip them for constructive dialogue with stakeholders outside the immediate development team.

Keywords Artificial intelligence (AI), Fairness, Algorithms, Jurgen Habermas, Domination-free development environment, Process model

*Correspondence:

Marco De Luca
marco.deluca@unina.it

¹ Department of Fair AI Development and Algorithmic Bias, Faculty of Engineering, University of Naples Federico II, Naples, Italy

² Department of Bias Mitigation and Developer Practices, Faculty of Engineering, University of Bologna, Bologna, Italy

³ Department of Responsible AI and Fairness, Faculty of Engineering, University of Florence, Florence, Florence

Introduction

Escalating apprehensions regarding the societal consequences of biased artificial intelligence (AI) have generated calls for a fundamental shift in how AI algorithms are engineered [1]. These anxieties underscore the necessity of close collaboration among AI developers, organizations, management, decision-makers, and the public sector that stands to be impacted by AI [2-4]. Although data quality and ethical reflection are vital elements in producing unbiased AI, the duty of engineering equitable AI and managing its ethical ramifications rests heavily upon AI developers [5]. Because developers design, train, observe, assess, and test AI algorithms, they bear responsibility for the moral outcomes of the choices generated by these systems [5-10]. AI models execute precisely what they are instructed to do, making AI developers pivotal actors in the realization of equitable AI [5, 9].

AI developers are increasingly urged to emphasize the social and moral dimensions of constructing equitable AI and to actively eliminate bias from the models they create [9]. However, the specific role of AI developers in mitigating algorithmic bias through equitable development methodologies remains largely unexamined. While substantial literature exists regarding the value of explainable AI—focusing on creating AI systems that are transparent and comprehensible to the public [11–13]—obstacles persist concerning how these explanations are practically applied and theorized [13]. A structured process model that encourages transparent dialogue and partnership between developers and AI stakeholders could be highly effective in generating fair AI. This approach can support explainable AI and foster the creation of models that command the understanding and trust of all parties, including the communities affected by them. It is vital that individuals impacted by algorithmic determinations are made aware of the situation, supplied with relevant data, and permitted to understand the underlying logic of those algorithms [14]. The manner in which enterprises build and implement technology fundamentally shapes public trust [6]. This dynamic is examined further under the training and development theme of this study, where developers highlight the necessity of transparency and explainability for achieving equitable AI.

Xivuri and Twinomurizi [15] formulated a process model grounded in Habermas' theory of communicative action to maintain adherence to fair practices throughout AI creation. This framework offers AI developers an equity-focused process model to prioritize ethical dimensions during AI design and execution, moving beyond a purely technological focus. It prompts them to evaluate the broader social impact of their work and actively solicit stakeholder input. Such a process model fosters ongoing refinement, thereby elevating the standard of AI algorithms. Nevertheless, despite the value of these process models, a gap remains in our understanding of how AI developers interact with them to generate fair AI.

Consequently, this study adopted an exploratory qualitative research approach to engage directly with AI developers, aiming to comprehend how they interface with equitable practices to produce fair AI. The investigation was designed to address the following central question: How can AI developers contribute to developing fair AI processes to ensure the fairness of AI algorithms?

The insights gained were subsequently applied to formulate practical parameters supporting the execution of the Habermas framework within equitable AI workflows.

This research contributes to actionable frameworks illustrating how AI developers can foster fair AI via equitable processes. The insights expose multiple areas of concern within the AI engineering workflow, such as an absence of gender and social diversity within development teams. Managers may prioritize hitting specific performance targets, occasionally bypassing vital validation steps. Because developers are frequently the sole individuals who fully grasp the technical work, a lack of personal integrity can lead them to hide biased outcomes from management and outer stakeholders. Additionally, while models undergo validation before deployment, representatives from the diverse social groups most vulnerable to the impacts are routinely left out of the testing cohorts.

Recommendations are offered across four domains: governance, social, technical, and training and development procedures. Responsible corporations must take explicit steps to ensure their technical teams abide by equitable frameworks during AI engineering; developers must foreground ethical priorities and the potential social ramifications of their models; alliances should be forged between AI developers, stakeholders, and the public, particularly those groups exposed to the consequences of AI systems; and AI developers must guarantee transparency and explainability within their architectures while executing rigorous testing protocols for bias and implementing remedies prior to deployment. Furthermore, emotional intelligence training should be integrated for developers to enable them to guide productive discussions with non-technical individuals, fostering mutual understanding and the creation of fair AI.

The subsequent sections of this paper are organized as follows: the following section outlines the theoretical context regarding AI algorithmic equity, alongside the Habermas framework for fair AI and its relevance for developers. This is

followed by an overview of the research methodology, an analysis of the insights, a summary of the discussion, limitations, practical implications, and a final conclusion.

AI Frameworks/Models (Theoretical Background)

AI algorithmic fairness

The deployment of discriminatory AI systems has resulted in numerous instances of inequitable outcomes, reinforcing historical societal imbalances. For instance, facial recognition software has faced severe criticism for exhibiting bias against specific racial minorities. Similarly, Amazon's automated hiring system was criticized for demonstrating bias against female candidates [16], while a risk-prediction tool utilized within the US legal system was shown to falsely categorize Black defendants as high-risk at a higher rate than white defendants [16, 17]. These cases exemplify biased AI and underscore the profound effect that these technologies can exert on specific social demographics. Achieving fairness in algorithmic decision-making has emerged as a paramount challenge for multiple industries and broader society [18, 19]. This reality underscores the necessity for AI developers to fully comprehend the social repercussions of biased AI systems.

Constructing equitable AI necessitates a domination-free engineering workflow and environment, where moral principles are emphasized throughout the entire development lifecycle. Habermas conceptualized a domination-free environment as a space clear of power asymmetries, designed to cultivate authentic comprehension, consensus, and democratic choice-making. Such an environment ensures that all involved parties possess an equal platform to articulate their perspectives, priorities, and anxieties free from intimidation or manipulation.

The following section presents a more comprehensive analysis of the Habermas framework and its specific utility for securing AI algorithmic fairness.

The Habermas approach

A Habermasian approach to fair processes in AI algorithmic fairness holds immense relevance for AI developers, who occupy the frontline of algorithmic construction. This framework stresses the necessity of participating in communication that is completely devoid of manipulation, while following a designated set of guidelines to eliminate prejudice and incorporate affected populations throughout every phase of creation [15]. The methodology is derived from the scholarship of Jürgen Habermas, an eminent philosopher and sociologist acclaimed for his theories regarding communicative action. Habermas highlights the importance of entering into transparent and all-inclusive dialogue to attain equitable decision-making frameworks. Applied to the domain of AI algorithmic fairness, this necessitates that creators proactively gather perspectives from a diverse array of stakeholders, establishing avenues for meaningful debate unencumbered by power asymmetries or coercion. This perspective centers specifically on adhering to three distinct communicative criteria: logical-semantic, procedural, and performative rules, each of which is vital for AI developers to execute. **Figure 1** illustrates the Habermas process framework designed for AI algorithmic fairness.

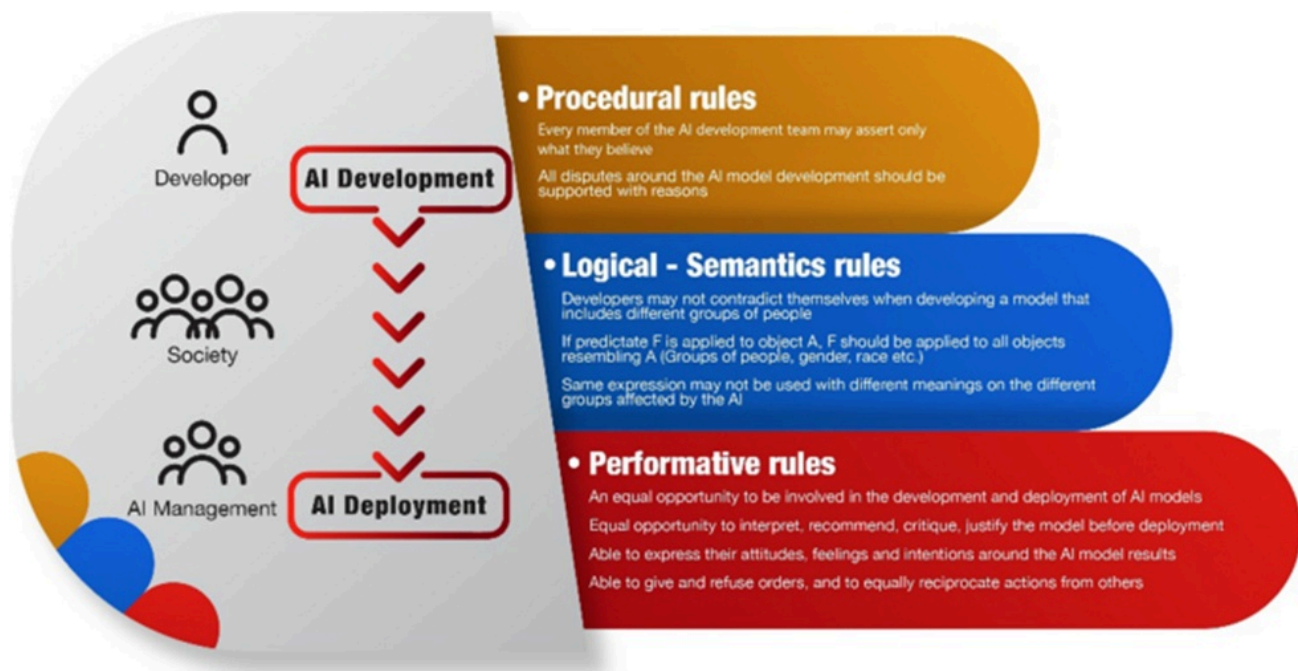


Figure 1. Process framework for AI algorithmic fairness—a Habermasian approach

By complying with logical-semantic principles, AI creators can foster a production setting that remains entirely free from coercion. For instance, they can ensure that parameters applied to entity A are identically applied to entity B, provided that both A and B denote equivalent categories (such as female/male).

Procedural principles demonstrate the way creators can establish an environment that enables the technical team to attain a deep comprehension by assessing claims of validity. For instance, team members can present and contest their viewpoints using validated justifications, which serves to reveal underlying prejudices and confirm the equity of the AI algorithm.

Performative principles illustrate how software creators can guarantee that their technical cohorts achieve mutual comprehension while maintaining an equilibrium of influence during the active cycles of AI production. This is vital for confirming that all interested parties are integrated into the architecture and that the resulting AI algorithm remains just and unbiased.

The subsequent section delineates the methodology utilized in this investigation.

Research Approach

This investigation utilized a qualitative research design to explore the ways in which AI developers can engage collaboratively within an equitable algorithmic development framework. An open-structured interview format was employed for the purpose of data collection. Ethical approval for gathering this data was granted by the institutional ethics board.

The research questions were structured to correspond directly with the Habermasian model of communicative engagements liberated from domination. Consequently, they were segmented into three primary domains: logical-semantic rules, procedural rules, and performative rules. Because the primary objective of the study was to consult directly with AI engineering professionals, purposive sampling was deployed. Purposive sampling is utilized to pinpoint specific research sites that possess distinct and required contextual attributes [20].

The verbal accounts gathered from these interviews were converted into text and interpreted through thematic analysis utilizing Atlas.ti to map out the collective understandings and shared professional realities of the developers [21]. More specifically, an abductive approach to data coding and interpretation was utilized to detect recurring patterns and themes, thereby deriving theoretical insights from the evaluated material [22]. Abductive interpretation and the critical theory of Jurgen Habermas share the mutual objective of scrutinizing societal challenges and exposing their deep-seated structures and origins. The coding process was systematically categorized under the three core tenets of communicative interaction: logical-semantic, procedural, and performative rules.

The ultimate goal of this data assessment was to ascertain whether a commitment to the Habermasian framework for equitable AI systems could effectively mitigate algorithmic bias.

Demographic analysis: AI developers

A total of ten (10) AI developers participated in the interview process. The specific background characteristics of these 10 professionals are detailed in Table 1 below. The dominant educational qualification represented was a PhD (30%), which was followed sequentially by Master's degrees (30%), Honours degrees (20%), and Bachelor's degrees (20%). Table 1 highlights a minimal representation of female AI engineering professionals, as this cohort proved exceptionally challenging to recruit despite the application of a snowball sampling approach.

Table 1. AI developers' demographics

Sex	Employment Sector	AI Experience (Years)	Educational Attainment
Male	Information and communications technology	12	Doctorate (PhD)
Male	Educational sector	5	Doctorate (PhD)
Male	Information and communications technology	12	Master's degree
Male	Wholesale and retail trade	4	Master's degree
Male	Information and communications technology	7	Honours degree
Male	Banking sector	6	Honours degree
Male	Banking sector	5	Bachelor's degree
Female	Agricultural sector	1	Doctorate (PhD)
Female	Telecommunications industry	5	Master's degree
Female	Information and communications technology	4	Bachelor's degree

The professional sectors of the participants were distributed across Information and Communication Technology (40%), Banking (20%), Telecommunications (10%), Retail (10%), Education (10%), Agriculture (10%), and Wholesale and Retail (10%), with their professional tenure in the AI space spanning from 1 to 12 years. The software engineers involved in the study were located in South Africa, a region that serves as a base for multiple multinational enterprises.

Content analysis: perspectives of AI developers on AI algorithmic fairness

The analysis yielded fifty-five (55) distinctive codes connected to logical-semantic rules, twenty-one (21) codes associated with procedural rules, and forty-one (41) codes tied to performative rules, culminating in an aggregate of 117 codes spanning the three communicative principles. This distribution is visually detailed in **Figure 2** below. These codes were further classified into an assortment of themes and sub-themes, uncovering 30 codes focused on Governance (encompassing AI guidelines, oversight frameworks, and accountable institutions), 27 codes tied to Social Responsibility (covering partnerships between software engineers, corporate management, subject matter experts, and the communities impacted by the models), 31 codes relating to Technical facets (involving data harvesting, validation routines, algorithm testing, and post-development inspection), and finally, 29 codes dedicated to Training and Development (addressing model interpretability, transparency, engineer upskilling, ethical AI instruction, and emotional intelligence cultivation) across the logical-semantic, procedural, and performative domains. **Figure 2** below provides a breakdown of these thematic groupings and code distributions for the three communicative prerequisites.

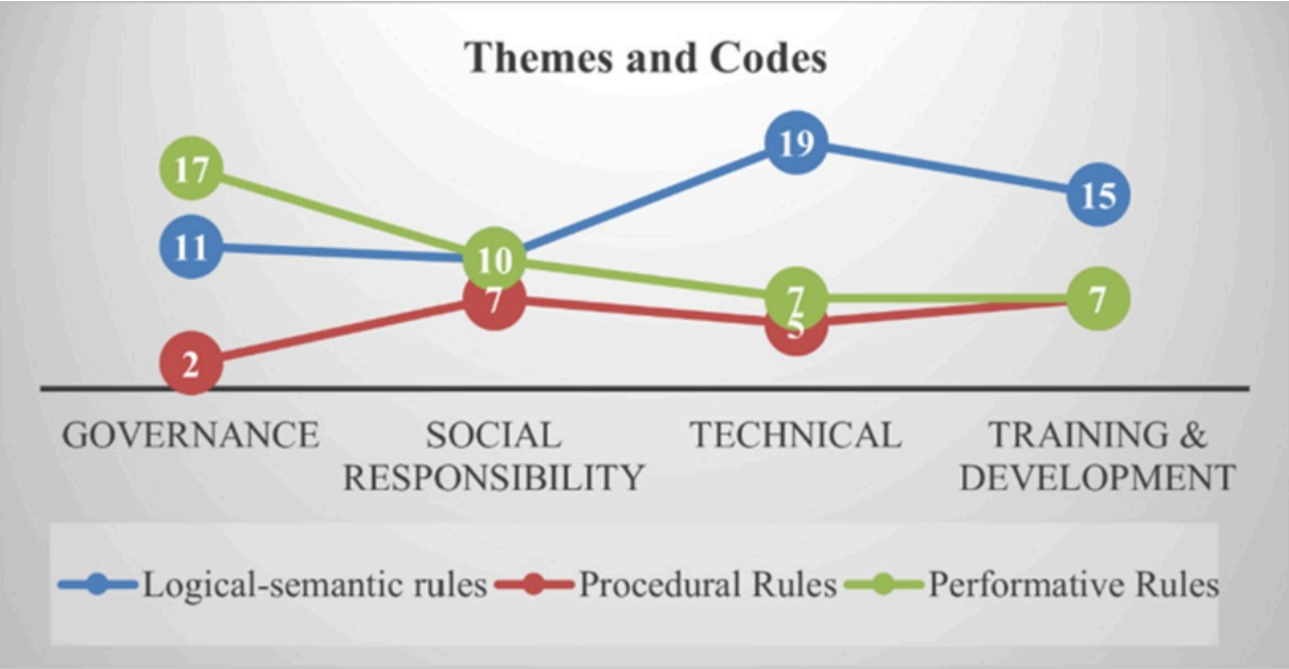


Figure 2. Identified themes and coding summary

Figure 2 above maps out the frequency with which specific thematic codes were mapped to the three distinct Habermasian communicative requirements. While a large portion of the themes find relevance across all three interactive criteria, there was a higher concentration of technical codes within the logical-semantic domain (19 codes), a greater density of social responsibility (7 codes) and training and development themes under procedural rules (7 codes), and a leading volume of governance-related designations within the performative rules domain (17 codes).

Results and Discussion

Drawing upon the empirical outcomes of this research, we examine the specific functions performed by software engineers within equitable development workflows structured around Habermas's communicative criteria. The investigation extracted the distinct thematic categories from the empirical data as detailed in **Table 2** below:

Table 2. The Habermas discursive rules and applicability of identified themes

Main Theme	Associated Subtheme	Logical–Semantic Principles	Procedural Principles	Performative Principles
Governance	AI policy and governance frameworks	✓	✓	✓
Governance	Organisational accountability	✓	✓	✓
Governance	Regulatory requirements for AI	✓	∅	✓
Social responsibility	Collaborative practices	✓	✓	✓
Social responsibility	Development team considerations	✓	✓	∅
Technical aspects	AI validation and testing	✓	✓	✓
Technical aspects	Data assessment and review	✓	✓	✓
Technical aspects	Model evaluation processes	∅	∅	✓
Training and capacity building	Explainability of AI systems	✓	✓	✓
Training and capacity building	Transparency measures	✓	✓	✓
Training and capacity building	Emotional intelligence education	∅	✓	✓
Training and capacity building	Training for AI developers	✓	∅	∅
Training and capacity building	AI ethics education and training	✓	∅	✓

Logical-semantic rules

Logical-semantic rules focus on the methods by which AI developers can prevent self-contradiction during the design of algorithmic models that impact diverse demographic segments. They further explore how engineering professionals can be upskilled to spot protected characteristics and understand their integration within a model. Additionally, logical-semantic rules evaluate the mechanisms through which training data can be audited to verify that it does not harbor prejudices against specific marginalized groups and remains truly representative of the target population. The themes relevant to logical-semantic rules comprise Governance, which manifests as AI policies and oversight structures (8 codes), responsible organizations (2 codes), and the demand for formalized AI oversight (1 code); Social Responsibility, observed through collaborative actions among software creators, leadership, industry experts, and impacted populations (9 codes), alongside the requirement for cross-functional development teams (1 code); Technical considerations, seen in data collection and auditing workflows (17 codes) and algorithm testing routines executed pre- and post-deployment (2 codes); and lastly, Training and Development, which includes the imperative for clear and interpretable models (12 codes), specialized developer upskilling (1 code), and ethical AI coursework (2 codes). **Figure 3** below illustrates the breakdown of themes and codes mapped to logical-semantic rules.

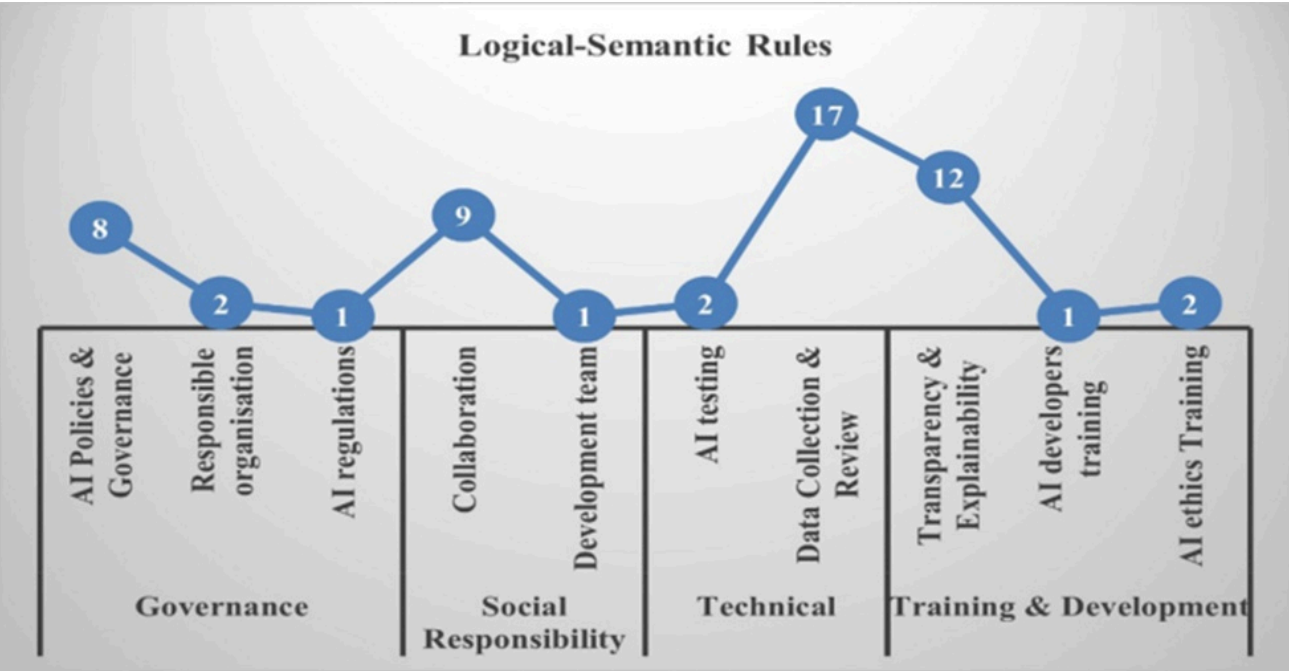


Figure 3. Logical-semantic rules—identified themes and coding summary

Governance

AI policies and governance

AI policies and governance demand the formulation of mandatory rules, oversight frameworks, institutional structures, and standard procedures that software engineers must follow to ensure the creation of ethical, non-discriminatory AI systems. Eight (8) distinct codes highlight the necessity for equitable AI strategies and governance structures aligned with industry benchmarks, the establishment of and compliance with an AI ethical charter, and the prerequisite that all AI initiatives undergo rigorous institutional and ethical vetting. As observed by two study participants:

“By obtaining approval from the ethical committee and an ethical clearance certificate, we may prevent having AI engineers contradict themselves while building a model that involves groups of people. A policy statement that formally describes the role of AI in the evolution of humanity is known as an AI code of ethics, also known as an AI value platform. The goal of an AI code of ethics is to advise stakeholders when faced with an ethical decision concerning the use of AI.” Developer 1.

“With primary data, the ethical committee needs to ensure that the data you are collecting doesn't violate certain ethics and principles. This committee is involved in the data collection and data analysis stage in ensuring that the data used is not biased in any way.” Developer 2.

The empirical data presented above points to a critical demand for launching explicit policy paradigms and compliance structures to guarantee the production of balanced AI models, provided these systems are strictly operationalized. Codified AI guidelines and structured administrative frameworks function as powerful mechanisms to guide software engineers toward building fair AI models [3].

Responsible organisation

An institution is deemed responsible when it takes proactive, intentional steps to orient its entire operational framework toward the creation and execution of equitable AI systems. Within this theme, the first of the two (2) identifying codes emphasized that businesses must accept liability, introduce governance frameworks, and establish ethical mandates for

algorithmic systems. Conversely, the second code concentrated on the vital need to cultivate public and societal confidence. This dual requirement was illustrated by a pair of participants:

“Enterprises ought to implement oversight frameworks and policies that explicitly dictate the parameters for managing personally identifiable data, maintaining rigorous compliance with protocols like POPI. Furthermore, systematic methodologies should be established to evaluate system outputs and performance, typically guided by a structured, predefined matrix.” Developer 9.

“The core requirement is establishing trust, which demands that enterprises actively manage their brand image, given that certain brands inherently lack public confidence. Once a solid brand reputation is established, corporations must foster clarity without diving into technical intricacies. Stakeholders simply require straightforward descriptions outlining the system's function, rather than granular insights into the underlying mathematics.” Developer 7.

AI regulations

AI regulations represent the governance structures and foundational principles instituted by state authorities to mandate that automated models are constructed ethically and without prejudice. The objective of these regulatory frameworks is to strike a balance between harnessing the innovative benefits of automation and mitigating its systemic hazards, thereby ensuring technological deployment mirrors human rights and public values. One (1) code highlighted the necessity for state entities to draft balanced compliance benchmarks and legal constraints. As one participant remarked:

“By aligning workflows with current statutory and compliance mandates, software engineers can successfully pinpoint protected characteristics and systematically prevent their inclusion during the training of AI models.” Developer 4.

Consequently, when definitive compliance frameworks for equitable automation are enforced, engineers are better equipped to recognize and omit protected characteristics that might otherwise introduce systemic bias into the resulting models.

Social responsibility

Collaboration

Collaboration signifies the obligation of engineering cohorts to ensure that external stakeholders, who exist outside the core technical implementation unit, comprehend the automated tools being deployed. Nine (9) distinct codes associated with this concept underscored the importance of translating specialized technical dynamics for non-specialist audiences, alongside integrating interdisciplinary working groups or domain experts to offer guidance throughout the lifecycle of the model. These codes also highlighted the need to embed the communities affected by the software directly into the design pipeline, creating a unified environment for multi-stakeholder engagement. As articulated by two respondents:

“An essential capability for developers is translating their specialized workflows for broader audiences, moving past technical jargon to convey core messages to non-specialists. Equally vital is demonstrating the practical utility of these models to show how they interface with everyday scenarios. End users will ultimately embrace an automated system when its functional value is visible, remaining largely indifferent to the underlying architecture.” Developer 3.

“It is vital to assemble interdisciplinary panels of experts to steer all choices regarding the architecture, iteration, and roll-out of ethical ML [machine learning] and AI systems, ensuring these units audit both current and future algorithmic features. Additionally, consumer perspectives should be woven directly into the ethical design process by leveraging customer advisory councils—selected from a diverse cross-section of the user base—throughout the product lifecycle to gather insights on core automated themes.” Developer 5.

While a significant portion of software engineers recognized the value of embedding civic perspectives into the pipeline and demystifying automated logic, certain respondents contended that the public generally pays attention to a system's internal framework only when it malfunctions. This perspective was captured by an interviewee:

"While this strategy is logistically feasible, its utility is constrained because end users and system engineers operate with entirely different focal points. For instance, when utilizing biometric facial scanners, my primary objective is seamless operation and baseline trust, not an evaluation of the underlying mechanics. Technical breakdowns do not remedy this issue; however, from an organizational leadership perspective, comprehensive model comprehension is vital. Executives bear the burden of mastering system logic prior to deployment, whereas the general public does not require this depth of information." Developer 7.

Development team

There is a distinct mandate for diversity within engineering cohorts, meaning that professionals from varied social backgrounds and analytical viewpoints must cooperate to build impartial AI software. One (1) code explicitly addressed this necessity for engineering team diversity. According to two of the participants:

"Engineering cohorts require demographic diversity, which introduces varied analytical viewpoints, expands our collective information pool, and engages a broader array of individuals. Rather than relying solely on top-down mandates from executives dictating specific software builds, we should solicit early design inputs from the communities destined to be impacted by these tools before engineering begins, thereby neutralizing hidden biases." Developer 6.

"Algorithmic bias predominantly mirrors systemic societal prejudices. To counteract this, our primary step must involve integrating representatives from diverse demographic sectors directly into engineering units—such as ensuring balanced male and female participation for gender-adjacent models. Software cohorts must maintain equitable representation, ensuring every demographic vulnerable to discrimination is explicitly included in the engineering group. This structural equilibrium must extend to executive leadership as well. Without addressing this fundamental imbalance, subjective distortions will persist; a homogenous male cohort designing tools for female users will inherently embed a masculine bias shaped by historical social norms. True equity in engineering access, combined with a diverse leadership cadre making final strategic assessments, will significantly alleviate this issue. Balanced representation is essential across the entire end-to-end lifecycle." Developer 7.

The qualitative findings reveal that both the engineering teams and leadership cadres currently suffer from a deficit in diverse demographic representation, which directly contributes to the creation of biased AI software.

Technical

Data collection and review

Data collection and review encompasses the operational procedures established for gathering source information, selecting specific datasets, and auditing the empirical inputs used during model optimization. Within this domain, seventeen (17) distinct codes highlighted the critical imperative to inspect, evaluate, test, and validate data assets prior to their integration into a training pipeline. Furthermore, ensuring data anonymization remains a primary requirement. Several software engineers emphasized the necessity of utilizing large-scale datasets that accurately mirror the demographics of the communities impacted by the resulting technology. There is likewise an evident need to decouple the duties of the personnel tasked with data curation from the engineers utilizing that data to train algorithmic systems. As observed by two study participants:

"Prior to managing the information, operations must commence with systematic data gathering and statistical sampling to designate target groups, ensuring they accurately mirror the intended deployment environment of the model. After

acquiring this information, an independent cohort must verify that the dataset is adequately representative—separate from the collection team—followed by a third distinct unit auditing the data to guarantee comprehensive representation. Introducing these specialized organizational roles is essential for future workflows.” Developer 7.

“Information must undergo sequential procedures, specifically partitioned into training, testing, and validation phases. Successfully navigating these stages, particularly validation, mitigates the risk of systematic bias. Furthermore, utilizing extensive datasets is imperative, as model performance scales with data volume.” Developer 3.

An individual participant stressed the importance of omitting protected characteristics within classification and decision-making architectures:

“While demographic variables hold utility in specific algorithmic contexts, they should be omitted from classification or decision-making models that determine service allocation based on system outputs. Excluding variables like gender or ethnicity prevents the architecture from replicating biased or stereotypical judgments tied to specific demographic groups.” Developer 8.

AI testing

AI model testing is defined as the systematic evaluation of algorithmic architectures to identify present or latent bias. Two (2) codes within this domain underscored the imperative of assessing and auditing AI systems for disparate impacts both prior to and following their integration into production environments. According to two interviewees:

“Mitigating algorithmic bias can be effectively achieved by auditing the system both pre- and post-implementation.” Developer 5.

“Rigorous model assessment and verification are essential. The system must undergo evaluation across isolated data segments as well as the aggregate dataset.” Developer 4.

Training and development

Transparency and explainability

Transparency and explainability denote the capacity to comprehend, trace, and audit the foundational datasets alongside the underlying logic driving algorithmic determinations. Twelve (12) codes categorized under this theme highlighted how specialized tools and interpretable architectures (such as tree-based models) can make these systems comprehensible to communities and organizational stakeholders. Technical practitioners must possess the skill to translate complex algorithmic mechanics into accessible, narrative-driven business insights for non-technical partners. As stated by two of the interviewees:

“Avoid dense technical jargon when delivering presentations to corporate executives. A superior approach involves translating statistical outputs into functional business solutions or accessible narratives that resonate more clearly with stakeholders.” Developer 9.

“Integrate ethical AI principles directly into the software engineering and product deployment pipelines. Treating these as independent procedures merely introduces operational friction and unnecessary complexity for development units. By embedding novel ML [machine learning] oversight protocols into the institutional compliance matrix, we reinforce our core ethical standards. Engineering teams are required to evaluate all ML products from an ethical perspective, critically reviewing data gathering, minimization strategies, procedural clarity, and core human values.” Developer 5.

Conversely, several engineers maintained that demystifying complex models for executive leadership and the public presents significant practical challenges and may prove ineffective. They argued that embedding ethical constraints directly into the initial AI design phase is a more viable strategy, fostering stakeholder trust organically without requiring granular explainability. According to one participant:

"The process is inherently complex because a vast majority of deployed systems operate as black-box architectures, making it exceptionally challenging to articulate the underlying decision-making mechanics to non-technical corporate partners." Developer 9.

AI developers' training

One (1) code classified under software engineer training emphasized the necessity of instructing developers to refine their professional capabilities. This educational focus guarantees that engineers construct automated architectures that are straightforward to comprehend. As noted by a respondent:

"Utilizing user-focused software applications can significantly demystify algorithmic logic for executive leadership and the general public. For instance, mobile applications are ubiquitous, enabling even young toddlers to navigate their interfaces effortlessly. Consequently, we can engineer our data systems to include a front-end interface that completely conceals the backend technical complexities. By presenting the public only with a simplified user experience, they can more readily appreciate the system's utility through everyday apps." Developer 2.

AI ethics training

Two (2) codes associated with ethical instruction in automation underscored the imperative for engineers to undergo moral training alongside learning methods for managing protected variables. According to two of the participants:

"Ethical integrity is fundamental; any automated tool under construction must undergo an equitable design trajectory to ensure identical inputs yield identical outcomes across different demographics. Software engineers require structured instruction in AI ethics to actively prevent the introduction of subjective distortions during the engineering lifecycle." Developer 2.

"Technical practitioners must receive formal instruction regarding the correct methodologies for managing and isolating protected attributes." Developer 2.

Summary

The preceding section examined the methodologies through which software engineers can cultivate an engineering ecosystem that safeguards parity, such as ensuring that modifications applied to entity A correspond precisely with adjustments made to entity B, provided they signify identical demographic classifications (e.g., male and female cohorts). The primary discoveries regarding logical-semantic regulations expose multiple deficiencies within contemporary AI design workflows. A prominent vulnerability is the absence of formal, pre-project protocols for securing ethical approval prior to launching automation initiatives.

Furthermore, the data indicates a systemic deficit in public trust toward artificial intelligence enterprises, coupled with a failure by these institutions to take proactive measures to earn public confidence regarding equitable engineering. The public typically remains indifferent to the internal mechanics of a model until a system failure occurs. Additionally, qualitative feedback demonstrates that contemporary engineering cohorts and leadership boards entirely lack demographic diversity. Finally, existing workflows suffer from inadequate data evaluation and verification protocols.

These outcomes highlight a clear mandate for structural governance, including formal AI policies, oversight frameworks, and conscientious organizations dedicated to engineering automation for societal benefit. Furthermore, adhering to state-

enforced AI guidelines and fulfilling social responsibilities by partnering with interdisciplinary specialists and impacted communities are vital components. Technical workflows—including pre- and post-deployment audits, structural model assessments, and consumer feedback loops—are equally critical. Lastly, institutional training programs that incorporate ethical education for engineers, combined with the transparency and interpretability of the resulting automated systems, remain essential.

Procedural rules

Procedural regulations analyze the establishment of an inclusive environment that empowers all AI stakeholders to articulate their perspectives and challenge any design choice they believe might introduce algorithmic bias. The conceptual themes relevant to procedural regulations include: Governance, which encompasses the mandate for AI frameworks and oversight policies (1 code) alongside conscientious institutional structures (1 code); Social Responsibility, realized through interdisciplinary communication between engineers and affected communities (6 codes) and the requirement for demographic diversity within engineering units (1 code); Technical Factors, comprising dataset review methodologies (3 codes) and system auditing both pre- and post-implementation (2 codes); and finally, Training and Development, which involves the demand for interpretable and transparent architectures (4 codes) alongside emotional intelligence instruction for software engineers (3 codes). **Figure 4** below outlines the conceptual themes and corresponding code counts for procedural regulations.



Figure 4. Procedural rules—identified themes and coding summary

Governance

AI policies and governance

It is imperative to institute corporate AI principles and regulatory frameworks that facilitate the inclusion of diverse stakeholders within the engineering lifecycle, enabling them to voice objections regarding potential algorithmic prejudice. One (1) code addressing the requirement for AI principles and regulatory frameworks highlighted the necessity of

standardized design workflows. These procedures should foster communication among corporate executives, domain specialists, and the public groups affected by automated systems, ultimately guaranteeing the implementation of principled models. As stated by an interviewee:

“Establish formal operational compliance criteria when launching automated systems; enterprises must adhere to a standardized blueprint that unifies production workflows while safeguarding the ethical integrity of the models.” Developer 5.

Responsible organisation

To realize equitable automation, commercial enterprises must accept greater accountability by prioritizing fair design and deployment practices. Cultivating a conscientious organizational culture that champions the necessity of building and launching unbiased AI is paramount. One (1) code connected to organizational accountability emphasized the importance of institutions prioritizing mitigation protocols to neutralize bias, alongside engaging a broad array of stakeholders to secure fair system outcomes. According to a participant:

“As automated tools are increasingly integrated into high-stakes operational workflows, enterprises must actively master the methodologies required to intercept and neutralize algorithmic prejudice.” Developer 5.

Social responsibility

Collaboration

Interdisciplinary communication among all involved parties is vital to achieve a holistic comprehension of an AI system's functionality and to guarantee the creation of equitable automation. Fostering an environment where participants feel authorized to highlight potential biases is a critical element of this trajectory. Six (6) codes focused on collaboration underscored the necessity of engaging affected communities to optimize model performance, aligning the objectives of developers with those of external stakeholders, and applying design thinking strategies specifically calibrated for AI engineering. As noted by two of the participants:

“Automated engineering inherently leaves room for discrepancies and subsequent optimization, which explains the reliance on predictive accuracy metrics within these models. An engineer can gain valuable insights from external observers; integrating outside feedback is highly beneficial as it refines the system's performance. Any interested party can participate because they evaluate the model from unique viewpoints and deconstruct the problem differently. When refining an existing system, you require input from a diverse array of stakeholders to leverage their distinct perspectives. Engaging separate stakeholder groups is absolutely essential.” Developer 3.

“Operational frameworks like design thinking are already established, and these methodologies can be pivotal because they integrate end users and stakeholders prior to finalizing any software build. Currently, design thinking is restricted to conventional product development; however, this exact framework could be adapted for ML models. Tailoring design thinking specifically to machine learning would systematically protect equity. Because design thinking brings immense value to product engineering as a whole, a parallel approach explicitly calibrated for ML would be highly effective.” Developer 7.

Development team

It is vital to assemble an engineering cohort that is receptive to the external insights and recommendations voiced by diverse AI stakeholders. One (1) individual code addressing the development team emphasized the requirement for software engineers to possess open-mindedness and a willingness to accept external critiques and user inputs, thereby optimizing system performance and advancing equitable AI architecture. As articulated by a respondent:

“Algorithmic distortion can easily stem from personal preconceptions held by engineers or corporate managers. Technical teams frequently search for patterns that validate their own expectations, which can skew the output metrics of the system. It is critical for all personnel to maintain complete objectivity when initiating model development.” Developer 4.

This participant feedback indicates that software engineers and corporate leadership may harbor preconceived mentalities that dictate their professional inclinations, potentially introducing systematic partiality into the engineering pipeline. To mitigate this risk, engineering units must consciously maintain an inclusive, adaptive posture toward external viewpoints and collaborative feedback.

Technical

Data collection and review

The integration of diverse stakeholder perspectives within the engineering lifecycle must commence during the initial stages of data harvesting and auditing. Three (3) codes categorized under data collection and review highlighted the necessity of employing impartial sampling methodologies and establishing rigorous protocols for selecting appropriate model datasets. Software engineers must acknowledge that the integrity of an automated model is fundamentally constrained by the integrity of its training inputs. As stated by an interviewee:

“Algorithmic bias can manifest through diverse mechanisms, such as flawed sampling routines that completely overlook substantial demographic sectors, or confirmation bias, wherein a data scientist selectively incorporates datasets that reinforce their personal worldview. Remedying this issue demands a profound comprehension of how machine bias originates. For instance, supervised learning—a primary paradigm of artificial intelligence—relies entirely on the rote processing of ingestion data. Through this supervised paradigm, a calibrated model determines outcomes for completely novel datasets. Operating under the 'garbage in, garbage out' axiom, the standard of an automated decision is strictly bound by the quality of the information it processes.” Developer 5.

Certain engineering professionals maintained that because a model's efficacy is entirely dependent on its training inputs, establishing multi-stakeholder collaborative environments will yield negligible benefits unless the underlying data assets are systematically cleansed and balanced first.

AI testing

Auditing automated architectures both pre- and post-deployment is essential for the successful detection and mitigation of algorithmic bias. Two (2) codes associated with system evaluation underscored the critical need to test models both prior to production and following live implementation. As observed by a participant:

“Execute rigorous system evaluations before and after live deployment. Testing artificial intelligence and machine learning architectures represents a primary methodology for intercepting algorithmic bias prior to releasing the software into production environments.” Developer 5.

Furthermore, one professional advocated for conducting systematic bias audits and internally sharing those equity metrics with external departments within the enterprise. This recommendation implies that empirical verification results are generally isolated and not disseminated across broader corporate units. According to the respondent:

“The primary objective must center on mapping existing biases and explicitly communicating these limitations when deploying our systems. Eradicating systemic bias entirely is a far more formidable challenge.” Developer 10.

Training and development

Transparency and explainability

It is paramount that engineers make their automated architectures fully transparent and interpretable to stakeholders operating outside the core technical unit. This clarity empowers external partners to evaluate, critique, and flag latent prejudices embedded within the software logic. Four (4) codes focused on transparency and explainability highlighted the importance of documenting internal system mechanics, utilized datasets, training histories, and decision-making pathways. This perspective was shared by two interviewees:

“Elevate operational clarity. Artificial intelligence is perpetually limited by the opaque nature of its internal processes. Deep learning architectures, for example, leverage complex neural networks inspired by biological structures to generate conclusions, yet the exact logical trajectory remains obscured. The transition toward interpretable AI seeks to demystify exactly how training data is processed and how the underlying algorithm functions.” Developer 5.

“Operational clarity is essential; data practitioners must never conceal any attributes of their datasets. The foundational pillars are transparency and a comprehensive grasp of the core problem. Once you fully diagnose the issue, you can chart the precise trajectory required to achieve the solution, allowing you to confidently validate and defend the system's outputs.” Developer 9.

Emotional intelligence training

To interface productively with external stakeholders and integrate their diverse concerns, software engineers must cultivate emotional intelligence. Targeted emotional intelligence training can empower technical professionals to regulate interpersonal dynamics, communicate transparently with non-technical partners, and engineer equitable automated systems. Three (3) codes linked to emotional intelligence instruction emphasized the necessity of training AI engineers in soft skills, interpersonal communication, and openness to external guidance. According to an interviewee:

“Engineers must remain adaptable; they need to welcome architectural adjustments and show a willingness to integrate external viewpoints. To cultivate this flexible mindset, developers must be trained not merely to be proficient programmers, but to exhibit professional maturity and emotional awareness. Technical teams require comprehensive training in interpersonal soft skills.” Developer 2.

The qualitative evidence above suggests that artificial intelligence developers frequently exhibit resistance when presented with external critiques or recommendations originating from outside their immediate technical teams.

Summary

The preceding section scrutinized the methodologies through which enterprises can establish an inclusive environment that permits all AI stakeholders, particularly the general public impacted by these architectures, to achieve a comprehensive understanding of the automated models undergoing engineering. Among the principal insights concerning procedural regulations, the analysis identified a deficiency in coherent policies and governance frameworks, alongside a failure by corporations to accept accountability for constructing equitable AI. Furthermore, the empirical feedback

demonstrates that software engineers exhibit resistance toward integrating external critiques and recommendations from alternative stakeholders.

Both technical practitioners and organizations maintain an opaque posture regarding the datasets they leverage, the methodologies used to train their models, and the internal logic driving automated decisions. These outcomes underscore a clear requirement for corporate AI policies, structural governance, and conscientious enterprises dedicated to the realization of unbiased AI, alongside active multi-stakeholder collaboration. Additionally, the algorithmic architectures and training datasets must remain transparent and interpretable, while software engineers must receive instruction in emotional intelligence to foster a receptive attitude toward external evaluations and diverse viewpoints.

Performative rules

Performative regulations analyze the mechanisms through which software engineers can guarantee the active engagement of all AI stakeholders, ensure that definitive mandates regarding the avoidance of systemic prejudice are issued across all parties, and confirm that developers systematically execute recommendations provided by diverse contributors. The conceptual themes applicable to performative regulations include: Governance, comprising corporate AI policies and governance structures (9 codes), statutory AI regulations (2 codes), and responsible organizational behavior (6 codes); Social Responsibility, represented by multi-stakeholder collaboration (10 codes); Technical Factors, encompassing algorithmic testing routines (4 codes), data review workflows (1 code), and structural model reviews (2 codes); and finally, Training and Development, consisting of emotional intelligence instruction (4 codes), transparency and explainability initiatives (2 codes), and ethical AI training (1 code). **Figure 5** outlines the distribution of themes and codes identified for performative regulations.



Figure 5. Performative Rules—Identified themes and coding summary

Governance

AI policies and governance

To cultivate an equitable distribution of influence during the engineering of fair automation, institutions must establish AI policies and governance frameworks that authorize and streamline the involvement of all demographics affected by the model. This structural inclusion serves to safeguard the development of unbiased systems. Nine (9) codes clustered under AI policies and governance highlighted the imperative to align governance structures, operational protocols, corporate policies, and foundational principles, alongside implementing an explicit AI fairness charter and formalized operational guidelines. There is an evident demand for dedicated AI principles management committees and the execution of punitive measures for non-compliance with established ethical standards. As noted by one participant:

“Establish an internal definition of fairness calibrated for your enterprise. Draft a standardized template for an AI fairness charter and instruct every department utilizing automated tools to complete it relative to their operational context. This is particularly crucial for business managers, line executives, and product owners to preserve algorithmic equity throughout the operational supply chain. Furthermore, mandate that external vendors providing products or services with embedded AI architectures complete this fairness charter and conform strictly to institutional equity guidelines.” Developer 5.

Responsible organisation

A conscientious enterprise introduces definitive measures to ensure that both software engineers and the public segments impacted by an algorithm achieve a shared comprehension during the development of unbiased AI. Six (6) codes linked to organizational responsibility underscored the necessity of deploying accountable AI governance mechanisms. Executive leadership must confirm that the application of automation remains consistent with core corporate principles, explicitly embedding this alignment into the executive management agenda. The institution's strategic methodology for ensuring algorithmic parity must be formally disseminated, and the empirical outcomes of fairness assessments should be shared across the broader organization. This was supported by two interviewees:

“Defensive risk-mitigation strategies must be systematically embedded directly into the engineering cycle.” Developer 4.

“We can circumvent this form of algorithmic prejudice by executing a model of conscientious AI governance. Several operational protocols are vital to realize this oversight: 1. Formulate foundational AI principles—the executive leadership cohort must achieve complete alignment through a structured set of automated guidelines. Corporate leaders must convene to evaluate AI deployment and verify its alignment with institutional values. 2. Architect a responsible AI governance matrix. Organizational ethics and technological tools are fundamentally interconnected and must be managed in tandem. The executive agenda must require innovation directors to disclose how automated systems are engineered and utilized within core operations. 3. Operationalize this oversight framework. This can be achieved by establishing a centralized point of contact and technical support. Clear communication regarding the verification routines across the AI lifecycle must be maintained, and relevant analytical findings must be systematically documented at the conclusion of each phase.” Developer 5.

The qualitative insights above reveal a systemic misalignment and lack of cohesion among contemporary AI governance architectures, institutional frameworks, operational policies, design processes, fundamental principles, fairness charters, and documented standards. The data further indicates a disconnect between the active deployment of AI and core organizational values.

AI regulations

Two (2) codes categorized under AI regulations emphasized the critical mandate for formalized compliance criteria and strict statutory law enforcement mechanisms. As articulated by an interviewee:

“Imposing formal legal mandates and financial penalties when algorithms exhibit bias is essential; there is an urgent need for more rigorous statutory enforcement. Currently, South Africa lacks legal enforcement mechanisms regarding deployed technological architectures. The faster these legislative frameworks are implemented, the more meticulously enterprises will operate. If the financial consequence of a penalty is severe, developers will exercise extreme caution during the engineering phase. When corporations recognize they face major financial liabilities for biased models, they will dedicate the necessary resources to guarantee their systems remain impartial.” Developer 6.

The preceding feedback demonstrates that there are currently no functional law enforcement mechanisms or statutory penalties designed to penalize the deployment of biased and inequitable AI models. Implementing these legislative measures could serve as an essential driver to force organizations to prioritize ethical AI design.

Social responsibility

Collaboration

Interdisciplinary communication and partnership among software engineers, organizational stakeholders, and the public groups vulnerable to algorithmic impact are vital for cultivating a shared comprehension regarding the design of unbiased AI. Ten (10) codes categorized under collaboration emphasized the necessity of embedding all relevant stakeholders across the complete AI lifecycle. This inclusive approach acts as a structural mechanism to optimize system performance and guarantee that the technology does not perpetuate discrimination against specific demographic cohorts. According to two of the study participants:

“I have conceptualized a structural requirement for systemic integration. The core challenge centers on how we unify this operational ecosystem—spanning from executive directors down to end users and localized communities—with the engineer functioning as the central intermediary. How do these distinct factions interface to guarantee the delivery of superior systems? This relates directly to mobile application frameworks. We require a specialized digital platform that permits a common individual to seamlessly interface with software engineers, executive management, and the broader public. Establishing that specific chain of communication is a fundamental prerequisite.” Developer 2.

“This can be operationalized through a structured project conduit. It is necessary to architect an engineering pipeline capable of accommodating every single phase of product optimization, thereby ensuring that every participant provides constructive input from initial empirical research through final system verification.” Developer 8.

One technical practitioner noted that within current workflows, comprehensive access across the various phases of artificial intelligence engineering is typically restricted to the project manager. External stakeholders are routinely excluded from or left uninformed about the underlying design cycle, limiting their exposure strictly to the finalized software application. As observed by the participant:

“An engineering initiative encompasses multiple distinct dimensions, including end-user coordination, general project management, software development, algorithmic optimization, and deployment operations. In a typical corporate setting, comprehensive access across these varied domains is restricted solely to the project manager.” Developer 10.

Technical

Data collection and review

One (1) code connected to data collection and review highlighted the critical obligation for software engineers to achieve a comprehensive understanding of existing data architectures prior to initiating model construction. According to an interviewee:

"Practitioners must thoroughly diagnose the nature of the information currently available. As a data scientist, it is vital to analyze the underlying raw data files because the information visible within the centralized database has already undergone decoding, restructuring, and truncation. Frequently, vital data points are omitted because the database administrators do not comprehend the operational utility of specific fields." Developer 9.

AI testing

Diverse stakeholders, including the civic communities affected by automated systems, must actively participate in the verification of AI architectures. Representatives from the specific demographic groups susceptible to algorithmic impacts should be integrated directly into the evaluation pipeline. Four (4) codes focused on AI testing underscored the importance of executing rigorous fairness audits both prior to deployment and following live release. These equity assessments must be finalized and their empirical metrics shared with the broader enterprise before a model is authorized for production. As articulated by two respondents:

"Audit algorithmic equity prior to launching any technological feature. Mandate that internal business units and external contractors execute and internally disseminate the findings of fairness outcome evaluations before any automated script is permitted to go live. Once you pinpoint which demographic cohorts face potential disparity due to data imbalances, simulate user interactions from those specific demographics and strictly monitor system behavior. Formally communicate your strategic methodology regarding algorithmic equity. Organize collaborative learning workshops focused on fairness outcomes alongside consumers and front-facing personnel to scrutinize the equity evaluations of any new or revised products and services. This practice is exceptionally vital for marketing divisions, public relations teams, and customer support staff." Developer 5.

"Systemic distortions within the model are rigorously audited, and if an architecture exhibits algorithmic bias, its probability of being transitioned into a live production environment is significantly minimized." Developer 9.

The findings above demonstrate that the empirical results of automated system evaluations are generally kept confidential and are not disseminated outside the core engineering unit. One participant advocated for the necessity of reviewing fairness metrics collaboratively with individuals outside the technical team. This recommendation involves sharing equity outcomes with external stakeholders who can offer novel perspectives and critical insights. Implementing this practice facilitates a broader diversity of analytical viewpoints, ensuring a more comprehensive and holistic assessment of the model's fairness.

Model review

Automated architectures must undergo systematic evaluation by diverse AI stakeholders, with the resulting feedback communicated back to the software engineers responsible for model maintenance. Two (2) codes linked to model review emphasized the value of statistical performance audits. Technical practitioners should receive direct feedback from the public regarding the equity and fairness of the system. As stated by one interviewee:

"Modern operations utilize highly sophisticated statistical monitoring tools to evaluate how specific products are performing in the commercial market. It is crucial to route these statistical diagnostics directly back to the software engineering cohorts. These analytical outputs clarify exactly why consumers engage with or reject specific automated models. Receiving data-driven statistical feedback regarding system performance can significantly help developers determine the optimal technical adjustments required moving forward." Developer 7.

Training and development

Transparency and explainability

The integration of openness and interpretability is vital for cultivating a shared comprehension regarding equitable AI design among engineers, organizational partners, and the communities susceptible to algorithmic impacts. Two (2) codes associated with this domain emphasized the requirement for absolute clarity regarding the data assets used during training, alongside the mandate to systematically document the entire machine learning lifecycle and its subsequent analytical breakthroughs. As articulated by two of the participants:

“The primary entities involved are clients or commercial units presenting a distinct problem that requires resolution. Another critical dimension involves the strategic guidance provided to corporate leaders when uncovering anomalies or insights that can systematically enhance their operations. It is equally essential to engage marketing and sales divisions throughout the journey, providing consistent operational updates to every stakeholder group. This process demands rigorous exploratory analysis and a deep comprehension of the data. Ultimately, the end user remains the foundational priority, making it imperative to thoroughly diagnose their needs and behaviors.” Developer 9.

“Systemic distortions within the model are rigorously audited, and if an architecture exhibits algorithmic bias, its probability of being transitioned into a live production environment is significantly minimized. However, data practitioners possess the technical capability to obscure model partiality, and if such systems are deployed, their real-world performance will be severely compromised. Ensuring an architecture is free from discrimination relies heavily on the personal ethics of the engineer. Because technical teams hold exclusive mastery over the code and understand the exact manipulation of variables, they can easily conceal structural flaws, selectively choosing which statistical metrics to share with executives and which to suppress. Frequently, a data scientist is fully aware of latent bias within their model, but corporate managers are driven entirely by short-term performance metrics and refuse to allocate the necessary time to neutralize these imbalances. Certain practitioners lack professional integrity and ignore the foundational rules learned during ethical AI training. To counteract this, leadership cadres must possess technical proficiency so they can adequately supervise data scientists rather than relinquishing total control and demanding only finalized outputs. When executives surrender all oversight, they abdicate their managerial responsibility, enabling technical teams to operate without accountability. Conversely, technically literate managers can foster a fluid, collaborative dialogue with engineering cohorts.” Developer 9.

The qualitative evidence above demonstrates that the individuals vulnerable to algorithmic decisions represent the most critical component of the lifecycle and must be actively embedded within the engineering process. Furthermore, the feedback indicates that goal-oriented corporate leaders occasionally fail to dedicate sufficient operational timelines for engineers to ensure their models are fully unbiased.

AI ethics training

Software engineers must receive structured education in algorithmic ethics to internalize the imperative of constructing equitable technology and to ensure they adhere to workflows that facilitate unbiased model development. One (1) code explicitly focused on the necessity of mandating ethical AI coursework for developers. One technical practitioner asserted that to guarantee the delivery of fair automation, equity parameters must be formally integrated into the project's technical specifications. As noted by two interviewees:

“In practical application, mandates to mitigate algorithmic bias will only be executed by technical teams if they are explicitly integrated into the initial scope of work. Clients must formally articulate these equity requirements within the project specifications. Within artificial intelligence projects, operational timelines and budgets rarely accommodate specialized verification tasks unless they are explicitly itemized in the contract.” Developer 10.

“When constructing a system from an enterprise perspective, your focus is strictly on building the tool, leaving you completely decoupled from its downstream societal effects. As an engineer, you might operate entirely in physical isolation within a basement environment, completely focused on the core coding process.” Developer 7.

The preceding findings indicate that a segment of the developer population lacks a comprehensive understanding of why safeguarding algorithmic equity is vital, suggesting these professionals would benefit substantially from targeted AI ethics training.

Emotional intelligence training

Establishing a shared comprehension and balancing the distribution of influence between technical teams and external stakeholders requires software engineers to undergo comprehensive emotional intelligence instruction. This educational baseline empowers developers to engage in constructive communication with non-technical partners, facilitating a unified approach to equitable AI engineering. Four (4) codes categorized under emotional intelligence training highlighted the critical need to instruct developers to be receptive to external critiques and design recommendations originating outside the core technical cohort. According to two of the participants:

“Coordinate targeted workshops designed to train technical personnel to constructively process external evaluations of their workflows. Specialized data scientists can be invited to share personal case studies where their software faced rigorous criticism, explaining how that external feedback ultimately helped them optimize their model's performance.” Developer 3.

“When engineering a model, the primary objective is to resolve a specific operational bottleneck. However, software developers frequently develop an excessive emotional attachment to their code after working on it for extended periods, causing them to view critiques as personal attacks. They must maintain perspective and recognize that these systems are built purely for problem-solving; external criticism and feature recommendations are intended solely to refine the architecture, not to offend the creator. Engineers must maintain an adaptable mindset and realize that all participants share a unified objective to resolve the core issue. Furthermore, they must embrace agility during model iteration, acknowledging that software engineering is not dogmatic and demands the flexibility to adjust when unexpected variables arise. Institutions should institute compulsory short courses or mandatory training modules focusing on agile collaboration, managing feedback, and maintaining a collective vision. Organizations must not assume that technical teams inherently possess these interpersonal capabilities; these soft skills must be explicitly taught, as a developer's exceptional coding proficiency does not guarantee adequate emotional intelligence.” Developer 6.

The participant feedback outlined above indicates that artificial intelligence engineers are traditionally accustomed to isolated workflows, which can render them less receptive to external interventions or active stakeholder integration during the development cycle.

Summary

Performative regulations analyze the structural mechanisms through which enterprises can ensure that diverse stakeholders achieve a shared comprehension and maintain an equitable balance of power throughout the active development phases of automation. A primary discovery within this domain reveals a profound lack of alignment and structural cohesion among contemporary AI governance architectures, operational frameworks, corporate policies, internal processes, design principles, fairness charters, and documented guidelines. The qualitative data further underscores a distinct disconnect between core institutional values and live algorithmic workflows.

Software engineers have historically operated in isolation, routinely excluding constructive design inputs from executive leadership or the public groups vulnerable to algorithmic impacts. Furthermore, while automated systems undergo verification routines, the empirical results of these tests are systematically withheld from individuals outside the core engineering unit. Currently, there are no statutory law enforcement mechanisms or financial liabilities attached to the deployment of biased AI.

Additionally, project executives occasionally prioritize rapid output targets over allocating adequate operational windows for developers to perform extensive validation and hyperparameter tuning (adjusting model settings to maximize performance)—both of which are indispensable for neutralizing algorithmic bias prior to production release. This operational structure can result in technical teams holding exclusive insight into the system's mechanics; if these practitioners lack professional integrity, they can easily mask unfavorable outcomes from executive boards and external stakeholders. Ultimately, the empirical insights emphasize an urgent mandate for well-aligned corporate AI policies and robust governance frameworks that can actively safeguard equitable model development. Concurrently, responsible enterprises must ensure absolute alignment between organizational values and technical mandates, while maintaining total transparency during algorithmic fairness evaluations by continually disseminating test outcomes across the broader institution and conducting recurrent audits to detect latent bias.

Discussion Summary

Software engineers occupy a pivotal position in the institutionalization of equitable automated procedures. It is incumbent upon these technical practitioners to accept accountability for evaluating the downstream societal repercussions of algorithmic bias and to guarantee the orchestration of impartial systems. Attaining equitable automation necessitates the structural integration of fair procedural frameworks across governance, societal interaction, technical execution, and professional instruction. To guarantee that engineering cohorts comply with these equitable workflows, enterprises must accept greater responsibility and implement deliberate operational mechanisms.

Establishing dedicated collaborative environments that permit fluid interaction among software engineers, corporate stakeholders, and the public groups vulnerable to algorithmic impact is a foundational prerequisite for equitable system optimization. Furthermore, technical teams must safeguard absolute transparency and interpretability within their models, adhering to rigorous pre-deployment verification protocols designed to isolate bias and execute necessary algorithmic corrections. Concurrently, delivering targeted emotional intelligence instruction to programming cohorts can empower developers to cultivate productive communication channels with non-technical personnel, directly supporting the generation of equitable technology.

Ultimately, systematic compliance with logical-semantic, procedural, and performative regulations serves to foster an engineering pipeline completely liberated from structural dominance. **Table 3** delineates the comprehensive array of interventions that can be deployed to institutionalize equitable workflows during the generation of unbiased automation. Technical practitioners must prioritize these operational strategies to champion ethical and inclusive technological advancement.

Table 3. Discussion summary

Domain	Recommended Practice	Description	Accountable Stakeholder(s)
Governance	AI policies and governance mechanisms supporting equitable AI	Establish governance arrangements and policy frameworks that facilitate the creation of fair and unbiased AI systems.	AI policymakers and AI management

Governance	Organisational responsibility for fair AI practices	Ensure organisations intentionally implement measures that require developers to adhere to fairness-oriented procedures during AI development.	AI policymakers and AI management
Governance	Regulatory frameworks governing AI ethics	Develop and enforce regulations addressing ethical principles and standards applicable to AI systems.	Government
Social Responsibility	Stakeholder collaboration	Promote cooperation among AI developers, relevant stakeholders, and members of society who may be affected by AI applications throughout the development process.	AI management, AI developers, and representatives of potentially affected communities
Social Responsibility	Diverse and inclusive development teams	Foster multidisciplinary teams that engage with external stakeholders to encourage shared understanding, balanced decision-making, and the advancement of fair AI deployment.	Organisations and AI management
Technical	Fairness-oriented AI testing	Conduct testing activities involving individuals or groups likely to be influenced by the AI system, both prior to and following implementation, to identify and reduce potential biases.	AI management, AI developers, and representatives of potentially affected communities
Technical	Representative data review and validation	Implement data collection and evaluation procedures to confirm that training data accurately reflect the relevant population and are assessed for possible biases.	AI management
Technical	Model evaluation and feedback processes	Establish review procedures that enable assessment of AI models and provide constructive feedback to the developers responsible for their creation.	AI management
Training and Development	Transparency and explainability initiatives	Ensure that AI models are understandable to stakeholders beyond the development team, allowing them to scrutinise, question, and report concerns regarding algorithmic bias.	AI management and AI developers
Training and Development	Emotional intelligence capacity building	Provide training that equips AI developers with interpersonal skills necessary for effective communication, collaboration, receptiveness to feedback, and open-minded engagement throughout AI development.	AI management and AI developers
Training and Development	Professional development for AI developers	Offer training opportunities aimed at strengthening developers' technical competencies and enhancing their effectiveness in AI development.	AI management and AI developers
Training and Development	Education in AI ethics	Deliver instruction on ethical principles and responsibilities that developers should observe	

during AI design and implementation to

Theoretical implications

Regarding its conceptual contributions, this inquiry introduces a comprehensive fair process model engineered to elevate the professional accountability of software developers in securing algorithmic parity throughout the engineering lifecycle. This process architecture is specifically configured to mitigate intensifying apprehensions surrounding latent algorithmic bias and its severe societal ramifications. The study resolves these structural concerns by providing a systemic framework designed to intercept and neutralize bias at every phase of the design trajectory.

Furthermore, the inquiry outlines actionable criteria to guide enterprises, technical directors, and software engineers in maintaining equity across the entire development pipeline. It underscores the critical requirement for fair procedural frameworks that guarantee systemic distortions are mapped and corrected prior to live product roll-out. By introducing logical-semantic, procedural, and performative mandates, the model functions to diminish power asymmetries, thereby preventing any isolated entity or practitioner from exerting arbitrary or unchecked influence over the architecture and implementation of automated systems.

By formalizing this procedural matrix and offering pragmatic benchmarks for algorithmic equity, the study enriches the broader scholarship of technological ethics and fairness. In the aggregate, the theoretical significance of this research resides in its architectural model and actionable directives, both of which enforce developer accountability for algorithmic parity throughout the software lifecycle. By dampening systematic bias and advancing parity, the research aims to optimize the ethical integrity and societal footprint of automated systems.

Practical implications

This research outlines actionable operational criteria for software engineering cohorts to maintain absolute procedural equity throughout the construction of automated systems. Through systematic consultation with technical practitioners, the study mapped the foundational interventions necessary to achieve equitable automation. The execution of our proposed framework and directives carries direct operational ramifications for corporate governance, social integration, technical execution, and professional cultivation.

Governance: Enterprises are required to document and operationalize formal AI policies and oversight frameworks configured to champion equitable software engineering. These corporate policies and governance structures serve as indispensable institutional safeguards during the generation of fair automated models. These frameworks ensure that algorithmic architectures are engineered and deployed in a manner that explicitly preserves equity, operational clarity, and corporate accountability as codified within the corporate bylaws. Institutions bear an absolute responsibility to introduce deliberate measures ensuring their engineering units conform to equitable workflows. Concurrently, state actors must draft and enforce statutory regulations regarding ethical automation principles and constraints, protecting the public while compelling organizations and engineers to design unbiased technology.

Social: Collaborative environments must be established to permit software engineers to interface seamlessly with corporate executives, system stakeholders, and the civic demographics vulnerable to downstream algorithmic impacts. These integrated environments should cultivate transparent communication, allowing all participating entities to achieve a shared comprehension and maintain balanced power dynamics, ultimately leading to the execution of equitable automation. Organizational leaders and technical directors must prioritize demographic and social diversity within their engineering cohorts.

Technical: Data harvesting and verification workflows must be institutionalized to ensure that the datasets leveraged during model training accurately mirror the broader demographics of the relevant population. Technical safeguards, including rigorous and comprehensive model evaluation routines, must be implemented with the active participation of representatives from diverse demographic cohorts. The empirical metrics derived from these fairness evaluations must be shared transparently with stakeholders operating outside the core technical unit. Finally, all automated models must undergo systemic peer reviews, with the resulting optimization feedback routed directly back to the software engineers tasked with model design.

Training and development: Software engineers must guarantee the operational transparency and interpretability of their models for individuals operating outside the immediate technical unit. Targeted emotional intelligence instruction should be delivered to programming cohorts to empower them to interface productively with interdisciplinary stakeholders across the development lifecycle, thereby safeguarding the engineering of equitable automated systems.

Limitations

The empirical findings indicate that a significant segment of the developer population fails to adequately conceptualize the broader societal footprint of their models, which subsequently induces a distinct disinterest in resolving the ethical ramifications of automation. Despite the clear challenges encountered when recruiting software engineers willing to participate in comprehensive dialogues regarding ethical product development, ten (10) technical practitioners were ultimately secured for formal interviews. This inquiry was structurally restricted to software engineers operating within a single national jurisdiction due to the logistical and legal complexities associated with securing cross-border ethical clearances. Future empirical investigations should evaluate adoption dynamics among software engineers across a more diverse array of countries and geographic regions.

Conclusion

This research set out to resolve the following core inquiry: How can software engineers actively contribute to the structure of equitable AI processes to guarantee the fairness of automated algorithms? Utilizing a qualitative methodology, the study analyzed how technical practitioners interface with a Habermasian communicative approach to preserve procedural equity throughout the engineering of automated systems.

The primary discoveries expose a severe deficit in gender and social diversity within contemporary engineering cohorts. Furthermore, corporate managers are routinely driven by intense organizational pressure to deliver long-awaited, high-stakes results, which frequently causes critical equity verifications to be bypassed. The personal honesty and professional credibility of software engineers represent a critical vulnerability, given that developers possess the technical capacity to mask algorithmic distortions from both executive boards and external partners. Finally, pre-deployment verification routines rarely incorporate an adequate representation of the diverse civic demographics vulnerable to downstream technological impacts.

The empirical data demonstrates a clear opportunity to transition from distorted automation to equitable systems by engineering fair procedural frameworks. Software engineers occupy a vital position within the technical lifecycle and must systematically prioritize the moral dimensions of automation throughout the design phase. Practitioners must comprehend the severe real-world impacts of biased technology on the public, maintaining an absolute commitment to building fair systems. The study outlines actionable directives for software engineers to preserve procedural equity during code construction, spanning governance, social cooperation, technical execution, and professional instruction.

It is paramount that enterprises engaged in automated engineering accept accountability and implement deliberate protocols to ensure their technical teams comply with these equitable workflows. This mandate requires the establishment of integrated communication environments where developers can interface directly with executive leadership, policy architects, and the civic communities vulnerable to algorithmic impact; the execution of robust technical defenses such as comprehensive pre-release model validation; and the active integration of impacted public cohorts within the verification process.

Software engineers must ensure that their systems remain transparent and explainable to individuals operating outside the core technical cohort, and emotional intelligence coursework must be delivered to developers to guarantee productive collaboration with interdisciplinary stakeholders throughout the engineering of fair automation.

This inquiry underscores the urgent need to operationalize formal AI policies and corporate governance structures that systematically champion equitable development. Multiple sovereign entities and regulatory bodies—including the European Union via the General Data Protection Regulation (GDPR) and the proposed EU AI Act (AIA), alongside state architectures in the United States, the United Kingdom, Canada, China, Singapore, France, and New Zealand—have initiated strategic measures to draft policies, standards, statutory boundaries, and governance frameworks designed to shield society from algorithmic bias. Nevertheless, it is critical to note that while substantive progress has been achieved, legislative work within this domain remains ongoing. Consequently, this study advocates for an intensified focus on a process-oriented fairness model that champions structural adaptability and continuous optimization, remaining agile and highly responsive to shifting institutional demands and technical challenges.

The inquiry reinforces the supreme importance of parity within automated product design, offering practical benchmarks for software engineers to ensure their architectures are built through a fair and principled trajectory. The empirical insights carry substantial ramifications for corporations developing automated systems, forcing them to assume a highly active role in ensuring their programmers adhere to equitable workflows. As automation continues to exert a profound and expanding influence across everyday society, it remains absolutely critical to guarantee that these tools are engineered in a manner that remains fair, balanced, and equitable for all demographics.

The process-oriented framework introduced within this research provides a significant advancement for the field of ethical automated development. The study delivers actionable guidelines enabling software engineers to collaborate productively with all relevant partners throughout the engineering lifecycle, allowing machine bias to be successfully mapped and mitigated prior to live release, and systematically supporting the comprehensive design and deployment of fair automation.

In conclusion, future scholarly research is urgently required to isolate and evaluate the precise role of the general public in attaining algorithmic fairness. Subsequent empirical inquiries should explicitly explore the specific methodologies through which software engineers can successfully guarantee algorithmic parity across distinct global regions and sovereign jurisdictions.

Acknowledgements

The success of this study would not have been possible without the participation of the AI developers who took the time to share their valuable insights and experiences. We would like to express our sincere gratitude to all the AI developers who participated in this study. Your contribution is greatly appreciated.

Conflict of interest

None

Financial support

None

Ethics statement

None

Received: 10 Dec 2023

Revised: 28 Jan 2024

Accepted: 27 Mar 2024

Published online: 15 September 2024

Rights and permissions

Open Access The author(s) retain copyright. This article is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. It may be shared and adapted for non-commercial purposes with appropriate attribution, an indication of changes, and distribution of adaptations under the same license. Third-party material may be subject to separate terms identified in its credit line. View the license at <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

References

- Kapatamoyo M, Ramos-Gil YT, Márquez Domínguez C. Algorithmic discrimination and responsibility: selected examples from the United States of America and South America. In: *Commun Comput Inf Sci*. Cham: Springer; 2019. p. 147-57.
https://doi.org/10.1007/978-3-030-32475-9_11.
- Baker-Brunnbauer J. Management perspective of ethics in artificial intelligence. *AI Ethics*. 2021;1:173-81.
<https://doi.org/10.1007/s43681-020-00022-3>.
- Hermann E. Leveraging artificial intelligence in marketing for social good—an ethical perspective. *J Bus Ethics*. 2022;179(1):43-61.
<https://doi.org/10.1007/s10551-021-04843-y>.
- Telkamp JB, Anderson MH. The implications of diverse human moral foundations for assessing the ethicality of artificial intelligence. *J Bus Ethics*. 2022;178(4):961-76.
<https://doi.org/10.1007/s10551-022-05057-6>.
- Martin K. Ethical implications and accountability of algorithms. *J Bus Ethics*. 2019;160(4):835-50.
<https://doi.org/10.1007/s10551-018-3921-3>.
- Martin K, Shilton K, Smith J. Business and the ethical implications of technology: introduction to the symposium. *J Bus Ethics*. 2019;160(2):307-17.
<https://doi.org/10.1007/s10551-019-04213-9>.
- Mujtaba DF, Mahapatra NR. Ethical considerations in AI-based recruitment. In: *Proceedings of the 2019 IEEE International Symposium on Technology and Society (ISTAS)*; 2019. p. 1-7.
<https://doi.org/10.1109/ISTAS48451.2019.8937920>.
- Pethig F, Kroenung J. Biased humans, (un)biased algorithms? *J Bus Ethics*. 2023;183:637-52.
<https://doi.org/10.1007/s10551-022-05071-8>.

Sullivan YW, Fosso Wamba S. Moral judgments in the age of artificial intelligence. *J Bus Ethics*. 2022;178(4):917-43.
<https://doi.org/10.1007/s10551-022-05053-w>.

Tóth Z, Caruana R, Gruber T, Loebbecke C. The dawn of the AI robots: towards a new framework of AI robot accountability. *J Bus Ethics*. 2022;178(4):895-916.
<https://doi.org/10.1007/s10551-022-05050-z>.

Colaner N. Is explainable artificial intelligence intrinsically valuable? *AI Soc*. 2022;37(1):231-8.
<https://doi.org/10.1007/s00146-021-01184-2>.

Hunkenschroer AL, Luetge C. Ethics of AI-enabled recruiting and selection: a review and research agenda. *J Bus Ethics*. 2022;178(4):977-1007.
<https://doi.org/10.1007/s10551-022-05049-6>.

Samek W, Müller KR. Towards explainable artificial intelligence. In: Samek W, Montavon G, Vedaldi A, Hansen LK, Müller KR, editors. *Explainable AI: interpreting, explaining and visualizing deep learning*. Cham: Springer; 2019.

Marmolejo-Ramos F, Workman T, Walker C, Lenihan D, Moulds S, Correa JC, et al. AI-powered narrative building for facilitating public participation and engagement. *Discov Artif Intell*. 2022;2:7.
<https://doi.org/10.1007/s44163-022-00023-7>.

Xivuri K, Twinomurizi H. A Habermasian approach to fair processes in AI algorithms. In: Jembere E, Gerber AJ, Viriri S, Pillay A, editors. *Artificial intelligence research*. Cham: Springer; 2022.
https://doi.org/10.1007/978-3-030-95070-5_22.

Hamilton IA. Amazon built an AI tool to hire people but had to shut it down because it was discriminating against women. *Business Insider* [Internet]. 2018 Oct 10 [cited 2021 Aug 27]. Available from: <https://www.businessinsider.co.za/amazon-built-ai-to-hire-people-discriminated-against-women-2018-10>

Angwin J, Larson J, Mattu S, Kirchner L. Machine bias. *ProPublica* [Internet]. 2016 May 23 [cited 2023 Apr 8]. Available from: <https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing>

John-Mathews JM, Cardon D, Balagué C. From reality to world: a critical perspective on AI fairness. *J Bus Ethics*. 2022;178(4):945-59.
<https://doi.org/10.1007/s10551-022-05055-8>.

Morse L, Teodorescu MHM, Awwad Y, Kane GC. Do the ends justify the means? Variation in the distributive and procedural fairness of machine learning algorithms. *J Bus Ethics*. 2021.
<https://doi.org/10.1007/s10551-021-04939-5>.

Bhattacharjee A. *Social science research: principles, methods, and practices*. 2012.
<https://doi.org/10.4324/9781315458090>.

Braun V, Clarke V. Thematic analysis. In: Cooper H, editor. *APA handbook of research methods in psychology: research designs: quantitative, qualitative, neuropsychological, and biological*. Vol 2. Washington (DC): American Psychological Association; 2012. p. 57-71.

Saunders MNK, Lewis P, Thornhill A. *Research methods for business students*. 8th ed. Harlow: Pearson; 2019.