

Justiça Algorítmica: Instrumentalização, Limites Conceituais e Desafios na Engenharia de Software

Lucas Rodrigues Valença¹, Ronnie de Souza Santos¹

¹Department of Electrical & Software Engineering – University of Calgary
Calgary – AB – Canada

{lucas.rodriguesvalen,ronnie.desouzasantos}@ucalgary.ca

Abstract. *This article describes ongoing research with the aim of understanding the concept of justice in the field of software engineering, the factors that underlie the creation and instrumentalization of these concepts, and the limitations faced by software engineering when applying them. The expansion of the field of study called “algorithmic fairness” fundamentally consists in the creation of mechanisms and procedures based on mathematical and formal procedures to conceptualize, evaluate and reduce biases and discrimination caused by algorithms. We conducted a systematic mapping in the context of justice in software engineering, comprising the metrics and definitions of algorithmic justice, as well as the procedures and techniques for fairer decision-making systems. We propose a discussion about the limitations that arise due to the understanding of justice as an attribute of software and the result of decision-making, as well as the influence that the field suffers from the construction of computational thinking, which is constantly developed around abstractions. Finally, we reflect on potential paths that could help us move beyond the limits of algorithmic justice.*

Resumo. *Este artigo descreve uma pesquisa em andamento com o objetivo de compreender o conceito de justiça no campo da engenharia de software, os fatores que fundamentam a criação e instrumentalização desses conceitos e as limitações enfrentadas pela engenharia de software ao aplicá-los. A expansão do campo de estudo denominado de “justiça algorítmica” consiste fundamentalmente na criação de mecanismos e procedimentos matemáticos e formais para conceituar, avaliar e reduzir vieses e discriminações causadas por algoritmos. Realizamos um mapeamento sistemático no contexto de justiça na engenharia de software, compreendendo as métricas e definições de justiça algorítmica, assim como os procedimentos e técnicas para sistemas de tomada de decisão mais justos. Propomos, então, uma discussão acerca das limitações que surgem devido à compreensão de justiça como um atributo de software e resultado de tomadas de decisões, assim como a influência que o campo sofre decorrente da construção do pensamento computacional, que constantemente é desenvolvido em torno de abstrações. Por fim, refletimos sobre possíveis caminhos que podem nos ajudar a superar os limites da justiça algorítmica.*

1. Introdução

A empresa de jornalismo Propublica publicou, em 2016, uma investigação sobre o *Correctional Offender Management Profiling (COMPAS)*, uma ferramenta de avaliação de

risco amplamente utilizada nos tribunais dos Estados Unidos, relatando que o sistema rotulava pessoas negras que não possuíam reincidência criminal como alto risco duas vezes mais do que pessoas brancas [Angwin et al. 2016]. Sistemas de vigilância digital também têm sido alvo de críticas, um caso notável ocorreu no Brasil: os sistemas de reconhecimento facial foram usados como dispositivo de vigilância durante o carnaval, embora não haja evidências de que esses sistemas diminuam significativamente a criminalidade ou que previnam crimes durante o evento brasileiro, resultando na prisão de pessoas inocentes [Alencar 2023].

Essa crescente consciência de discriminação provocada por algoritmos levou a um crescimento de estudos sobre impactos sociais negativos de máquinas, sistemas digitais e algoritmos, que muitas vezes se estabelecem em conceitos como “*fairness*”, ou justiça [Verma and Rubin 2018]. Da mesma forma, a comunidade de pesquisadores da ciência da computação e engenharia de software tem proposto diferentes abordagens para testar quão justo é um sistema [Chen et al. 2024, Udeshi et al. 2018]. [Chen et al. 2024] define “bugs de justiça” como “qualquer imperfeição em um sistema de software que cause uma discrepância entre as condições de justiça existentes e necessárias” e “teste de justiça” como “qualquer atividade projetada para revelar *bugs* de justiça através da execução de código”.

A maior parte desses mecanismos se baseia em procedimentos matemáticos e formais para conceituar, avaliar e reduzir vieses e discriminações causadas por algoritmos. Porém, a maneira como algoritmos integram a sociedade, assim como os objetivos daqueles que os idealizam e desenvolvem e seus impactos, se entrelaçam em dinâmicas sociais que não são estruturadas de maneira estável. A partir dessa compreensão, esta pesquisa se propõe a analisar as limitações das abordagens de justiça algorítmica promovidas no campo da ciência da computação e engenharia de software.

2. Método

2.1. Objetivo da pesquisa

Este artigo faz parte de uma dissertação de mestrado em andamento. Como primeira etapa da pesquisa, está sendo realizado um estudo terciário sobre o conceito de justiça no campo da engenharia de software, os fatores que fundamentam a criação e a instrumentalização desses conceitos e as limitações enfrentadas pela engenharia de software ao aplicá-los.

2.2. Análise

Neste trabalho, realizamos um mapeamento sistemático terciário em conjunto com uma metassíntese. Metassínteses são definidas como as traduções interpretativas derivadas da integração ou comparação de descobertas entre estudos qualitativos [Cavalcante and de Oliveira 2020, Flick 2014]. Como explica [Stern 1997], a metassíntese visa revisar um grupo de estudos “com o objetivo de descobrir os elementos essenciais e traduzir os resultados em um produto final que transforme os resultados originais em uma nova conceituação”. O estudo é considerado um mapeamento terciário porque mapeia apenas revisões sistemáticas de literatura [Kitchenham et al. 2010]. Por fazer parte de uma pesquisa em andamento, aqui conduzimos apenas um mapeamento sistemático dos 10 primeiros artigos ¹, de um total de 60 incluídos na busca realizada, o

¹Os artigos podem ser encontrados no link a seguir: <https://figshare.com/s/01b30a1a7737696b5c2f>

que nos permite encontrar evidências que direcionem o foco da revisão sistemática que será realizada na dissertação [Keele et al. 2007].

2.3. Estratégia de busca

O processo de busca ocorreu de maneira manual e automática. Para a busca automática, uma *string* de busca² foi utilizada nos seguintes repositórios: *Google Scholar*, *ACM Digital Library*, *IEEE Xplore*, *Scopus* e *Sage Journals*. As fontes incluídas na busca manual estão presentes na Tabela 1.

Categoria	Fonte
Conferência	Foundations of Software Engineering (FSE)
Conferência	Empirical Software Engineering and Measurement (ESEM)
Conferência	FAccT
Conferência	ICSE: Research Papers
Conferência	ICSE: Software Engineering in Society
Conferência	ICSE: CHASE (Cooperative and Human Aspects on Software Engineering)
Conferência	ICSE: FairWare (Software Fairness)
Conferência	ICSE: GE (Gender Equality in Software Engineering)
Conferência	ICSE: NIER (New Ideas and Emerging Results Track)
Conferência	ICSE: RAIE (Responsible AI Engineering)
Conferência	ICSE: SE4RAI (Software Engineering for Responsible AI)
Periódico	ACM Transactions on Software Engineering and Methodology (TOSEM)
Periódico	Empirical Software Engineering Journal Issues
Periódico	Information and Software Technology

Tabela 1. Lista de fontes da busca manual

2.4. Processo de inclusão e exclusão

Os artigos incluídos (IC1) consistem em revisões de literatura que abordam definições de justiça em engenharia de software, incluindo, mas não se limitando a, fatores que fundamentam essas definições, como essas definições se relacionam com a discriminação algorítmica e os critérios usados para avaliar a justiça. Excluímos estudos que (EC1) não conduzem uma revisão de literatura de justiça de software ou testes de justiça, (EC2) não podem ser acessados através do proxy da Universidade de Calgary, (EC3) não são conduzidos em inglês, (EC4) são duplicatas - mantendo apenas a versão mais recente ou, no caso de um artigo de conferência e sua extensão de periódico, incluindo apenas o último - (EC5) não estão dentro do campo de engenharia de software e (EC6) foram publicados antes de 2017. Ao término da aplicação dos critérios, foram incluídos 60 artigos.

3. Resultados

3.1. Definições de justiça algorítmica

Muitos estudos têm focado em compreender o conceito de justiça no contexto de Inteligência Artificial (IA) [Verma and Rubin 2018, Soremekun et al. 2022, Pham et al. 2023, Rabonato and Berton 2024]. Nesse contexto, a noção de justiça é conceituada como a ausência de viés ou discriminação em decisões tomadas por sistemas

²(“software fairness” OR “fairness in AI” OR “fairness in ML” OR “algorithmic discrimination” OR “algorithmic bias” OR “software discrimination” OR “software bias” OR “fairness in software engineering”) AND (“systematic review” OR “systematic literature review” OR “literature review” OR “survey” OR “mapping study”) AND (“software engineering”)

digitais [Trigo et al. 2024]. Justiça algorítmica, ou justiça de software, é classificada como uma propriedade de um software [Chen et al. 2024] que busca garantir resultados, ou *outputs* que não discriminem grupos ou indivíduos [Trigo et al. 2024]. Como algoritmos são construídos em termos matemáticos e justiça é conceituada em função destes algoritmos, justiça também se torna um componente matemático e estatístico [Rabonato and Berton 2024, Palumbo et al. 2024], e suas definições e métricas se tornam frequentemente desenvolvidas a partir de uma compreensão de justiça procedural ou baseada na distribuição de bens e resultados.

Justiça procedural consiste no princípio de que todos devem ser tratados iguais durante um processo de decisão [Schwöbel and Remmers 2022]. Nesse sentido, *fairness through unawareness*, ou justiça através do desconhecimento, considera que um processo de tomada de decisão é justo se não considera atributos sensíveis (como raça e gênero) [Schwöbel and Remmers 2022, Chen et al. 2024]. Em contraponto, *fairness through awareness*, ou justiça através do conhecimento, considera que um sistema é justo caso ele classifique ou tome decisões iguais para indivíduos similares em relação a uma tarefa em particular [Dwork et al. 2011]. Dessa maneira, indivíduos considerados similares por um sistema de diagnóstico médico precisam receber o mesmo diagnóstico. A ideia de *fairness through awareness* consiste, então, na concepção de justiça baseada na distribuição de bens e resultados.

Os exemplos analisados acima são fundamentados na compreensão da justiça em nível individual. Em contraponto, justiça em nível de grupo consiste em avaliar as distribuições estatísticas entre diferentes grupos, buscando garantir paridade estatística em tomadas de decisões realizadas em diferentes grupos (homens e mulheres, por exemplo) [Dwork et al. 2011].

Débito de justiça, ou *fairness debt*, é definido como um conjunto de construções de design, implementação ou gestão que oferecem benefícios de curto prazo, mas estabelecem um contexto técnico que pode tornar mudanças no sistema custosas, resultando em impactos sociais significativos [de Souza Santos et al. 2025]. Aqui, no entanto, a abordagem sobre justiça algorítmica foca mais em suas causas — como viés histórico — e seus resultados, como danos psicológicos.

3.2. Métricas de justiça algorítmica

Diversas formulações são derivadas da concepção de justiça a nível individual e a nível de grupo, constantemente utilizando métricas como verdadeiro positivo, falso positivo, verdadeiro negativo e falso negativo [Dwork et al. 2011]. Paridade preditiva, por exemplo, exige que diferentes grupos tenham a mesma taxa de valor preditivo positivo (VPP). Ou seja, que diferentes grupos tenham, entre si, a mesma porcentagem de casos verdadeiros positivos em relação à quantidade total de casos positivos previstos [Verma and Rubin 2018]. As definições e métricas de justiça algorítmica são extensas e não se limitam aos exemplos trazidos, mas aqui propomos apenas uma compreensão dos conceitos que fundamentam o campo, visto que outros trabalhos já revisaram extensivamente as múltiplas definições de justiça algorítmica [Pham et al. 2023, Verma and Rubin 2018, Dwork et al. 2011, Soremekun et al. 2022, Rabonato and Berton 2024, Trigo et al. 2024].

A justiça algorítmica, entendida como um atributo do software, leva à criação do

teste de justiça. Esse teste envolve qualquer atividade destinada a identificar *bugs* de justiça — imperfeições em um sistema de software que resultam em discrepâncias entre as condições de justiça esperadas e as observadas — por meio da execução do código [Chen et al. 2024]. No contexto do aprendizado de máquina, [Chen et al. 2024] mapeia e categoriza os testes em cinco tipos: teste de dados, que busca identificar vieses nos dados de treinamento; teste do programa de aprendizado de máquina, que avalia falhas no código que podem comprometer a justiça; teste de *frameworks*, voltado para a detecção de falhas estruturais em bibliotecas de aprendizado de máquina; teste do modelo, que verifica o comportamento do modelo com base em suas entradas e saídas; e teste de componentes que não envolvem aprendizado de máquina, que investiga como elementos externos, como armazenamento de dados, podem introduzir vieses no sistema.

3.3. Procedimentos e técnicas para sistemas de tomada de decisão mais justos

Tendo definições e métricas de justiça algorítmica estruturadas, os mecanismos para instrumentalizar são classificados em abordagens de pré-processamento, processamento e pós-processamento [Caton and Haas 2024]. Abordagens de pré-processamento focam principalmente na instrumentalização de datasets. Datasets desbalanceados são constantemente compreendidos como fatores significativos no processo de discriminação algorítmica, daí torna-se fundamental a construção de datasets compostos por amostras representativas que reflitam de forma equilibrada as características de diferentes grupos sociais, minimizando vieses e promovendo uma distribuição mais justa dos resultados preditivos. Outras abordagens de pré-processamento consistem em ajustar os pesos dos dados (i.e. a importância de cada dado para a determinação do resultado final) [Kamiran and Calders] e realizar transformações probabilísticas nos dados para reduzir a influência de variáveis como gênero e raça nas previsões de modelos [Calmon et al. 2017].

Abordagens de processamento focam no algoritmo desenvolvido. *Adversarial learning* [Zhang et al. 2018] é uma técnica em que dois modelos são treinados simultaneamente. O primeiro modelo fica responsável pelas tomadas de decisões e o segundo tenta descobrir características sensíveis, como gênero ou raça. O objetivo é otimizar o modelo responsável pela tomada de decisão e minimizar a capacidade do segundo modelo, adversário, de prever características sensíveis, que podem influenciar em tomadas de decisões discriminatórias. Nessa abordagem, também é possível ajustar algoritmos de tomada de decisão adicionando restrições que se relacionam com as métricas de justiça algorítmica [Wu and He 2022]. Por exemplo, restrições que garantem que um modelo tenha a mesma taxa de verdadeiros positivos para diferentes grupos, ou que a diferença entre verdadeiros positivos e falsos positivos seja igual entre diferentes grupos [Rabonato and Berton 2024].

Abordagens de pós-processamento consistem em ajustar as decisões resultantes do modelo, que já foi treinado e executado. De maneira similar à técnica de restrição mencionada previamente, [Hardt et al. 2016] propõe ajustar previsões geradas através da aplicação de algoritmos de otimização, gerando novas previsões em que taxas de falso positivo e falso negativo sejam iguais entre grupos sensíveis. Ferramentas como *FairML* [Adebayo 2016] contemplam diversas métricas de justiça algorítmica, algoritmos de mitigação de vieses e técnicas para diagnosticar vieses em sistemas de tomada de decisão.

4. Discussão

4.1. Justiça como um atributo de *software*

A concepção de justiça algorítmica como um atributo de software [Soremekun et al. 2022] impõe alguns limites em relação às possibilidades de instrumentalização do conceito. Na engenharia de software, justiça é originalmente concebida como uma noção abstrata [Schwöbel and Remmers 2022]. Isso ocorre porque as dinâmicas de sistemas e algoritmos, ao se manifestarem na sociedade, escapam às representações próprias da engenharia de software. Para que o conceito se adeque aos processos da engenharia de software, este então é adaptado para uma noção formal, matemática e quantificável [Rabonato and Berton 2024] de justiça.

O primeiro erro então é estabelecido: a necessidade de adaptar os impactos sociais de algoritmos aos procedimentos formais da engenharia de software. Tal adaptação, na verdade, distorce a lógica esperada: a engenharia de software, que deveria adaptar-se às dinâmicas sociais que são produzidas através da interação entre seus sistemas e a sociedade, acaba articulando o campo de estudo em questão de maneira isolada, não reconhecendo que esta está em função da sociedade, e não o oposto. O que passa despercebido é que a própria formalização dos impactos sociais de algoritmos e sistemas digitais já é, por si só, uma intervenção da engenharia de software na sociedade. É a própria necessidade de formalizar os impactos sociais de algoritmos que produz a ideia de “justiça”: essa noção já nasce com um caráter limitante porque tem como elemento fundacional as tomadas de decisão realizadas por sistemas algorítmicos. Porém, os impactos sociais de algoritmos não se limitam às suas tomadas de decisão e distribuição de resultados.

Tomemos como exemplo o aplicativo de caronas Uber: em 2024, motoristas e passageiros foram assassinados durante corridas no Rio de Janeiro, porque o aplicativo sugeriu rotas que passavam por comunidades dominadas por traficantes [Danilo Vieira and Mariana Cardoso 2024, Roberta de Souza 2024]. Uma análise através das lentes de justiça algorítmica não nos permite investigar os impactos e as dinâmicas sociais estimuladas por algoritmos de rotas em aplicativos de corrida, quando contextualizados em relação à segurança pública da cidade.

4.2. Justiça como um resultado de tomadas de decisões

O segundo erro consiste então em definir os impactos sociais de algoritmos unicamente a partir de suas tomadas de decisão e distribuição de resultados, pois essa definição falha em reconhecer as características estruturantes de tecnologias na sociedade. Tecnologias de reconhecimento de gênero, como o *genderize*³, pressupõem a noção binária de gênero, e simplesmente avaliar quão justas são as predições desses sistemas “permite a ignorância das maneiras pelas quais os humanos e a tecnologia co-conspiram para [...] defender e reproduzir ativamente estruturas sociais discriminatórias” [Hoffmann 2019].

Os impactos e a discriminação algorítmica estão então estritamente relacionados não apenas às suas tomadas de decisões, mas à própria concepção das tecnologias desenvolvidas, e as abordagens de justiça propostas não apenas falham em mitigar os efeitos das estruturas maiores nas quais os sistemas algorítmicos são implantados [West 2020],

³*Genderize* é uma plataforma que utiliza inteligência artificial para supor o gênero atribuído à um nome, de acordo com o país escolhido.

como continuam a reproduzi-los. A noção de justiça a partir de resultados de tomadas de decisões significa também que esta se concentra em analisar quais grupos ou indivíduos estão em desvantagem devido às tomadas de decisão, e como [Hoffmann 2019] expõe, a mudança é sutil, mas com consequências: ao focar em uma análise sobre desvantagens, as abordagens de justiça algorítmica não questionam as condições normativas que produzem sujeitos favorecidos.

E mesmo que a definição de justiça algorítmica pudesse ser vista como um produto das decisões tomadas por um sistema, a engenharia de software ainda falharia pela maneira que olha para categorias estruturantes na sociedade, como raça e gênero: pesquisadores constantemente evitam justificar e explicar como adotam diferentes categorias de raça em estudos sobre justiça algorítmica, e que restrições técnicas e os próprios desejos dos pesquisadores levam-nos a esquemas de categorização mais simples com menos categorias raciais [Abdu et al. 2023]. Porém:

[...] os cientistas da computação não podem escapar do complicado trabalho político de classificação racial ao usar uma definição matemática particular de justiça. [Abdu et al. 2023]

4.3. O pensamento computacional e o papel da abstração

[Palumbo et al. 2024] afirma que princípios éticos, devido à sua subjetividade, são de pouca utilidade para cientistas da computação, pois não é possível encontrar maneiras de padronizá-los. A complexidade subjetiva com que algoritmos impactam dinâmicas sociais não é, no entanto, algo a ser combatido e eliminado. Na verdade, esse é o aspecto crucial que deve ser reconhecido caso a computação realmente busque se tornar uma ciência mais consciente dos seus impactos e mais comprometida com questões sociais. Porém, o caminho que parece ser seguido — a necessidade de formalizar matematicamente a justiça — justificada por uma suposta abstração presente no termo [Palumbo et al. 2024, Schwöbel and Remmers 2022], se contradiz quando justaposta a uma característica chave da computação: o pensamento computacional tem como uma de suas características a abstração. Para simplificar a complexidade de sistemas e tornar a programação mais eficiente, construímos nossas lógicas a partir da abstração de componentes que fazem parte destes próprios sistemas. Esse processo de abstração, porém, não consiste apenas em generalizar regras e construir métodos formais, mas também significa desenvolver novas maneiras de mapear realidades sociais, consequentemente determinando o que deve ser intervindo, e de que maneira [Malazita and Resetar 2019].

Não é verdade, então, que a justiça algorítmica nasce como um conceito abstrato, e por isso precisamos formalizá-la. Na verdade, o completo oposto ocorre a partir da seguinte contradição: como já mencionado, a engenharia de software, na tentativa de instrumentalizar os impactos sociais de algoritmos e sistemas digitais, formaliza esses impactos em termos de uma justiça baseada em tomadas de decisões probabilísticas, e é esse exato processo que desconsidera as dinâmicas sociais de algoritmos que surgem quando estes são imersos em estruturas sociais maiores. O maior desafio é compreender que, enquanto buscamos debater os impactos sociais dos algoritmos e aproximar a engenharia de software da sociedade, acabamos perpetuando maneiras de pensar o mundo que, sob a aparência de “justiça algorítmica”, continuam nos distanciando da sociedade e reforçando uma construção centrada em nossos próprios termos.

4.4. Possíveis caminhos

As problemáticas discutidas demonstram alguns dos sintomas de um campo de estudo mais próximo da ética empresarial convencional do que de tradições políticas e sociais mais radicais [Greene et al. 2019]. Não há uma alternativa intuitiva para isso, mas os exemplos da Uber e *genderize* permitem que compreendamos que um sistema supostamente justo não necessariamente reflete em impactos e dinâmicas sociais positivas. Talvez seja necessário, então, não apenas questionar se um sistema é justo, mas também de que maneiras ele interage com estruturas sociais maiores e resulta em dinâmicas sociais complexas e prejudiciais. E esse, claro, não é um problema que a computação está apta a resolver por si só: pesquisadores das ciências sociais, comunicação, direito e outras áreas têm constantemente buscado compreender essa questão. [Mendonca et al. 2023], por exemplo, concebe algoritmos como instituições nas sociedades contemporâneas, argumentando que eles não apenas estruturam a tomada de decisões, mas também constroem relações de poder.

Os impactos sociais dos sistemas algorítmicos não podem ser totalmente compreendidos a menos que sejam conceituados dentro de contextos institucionais e sistemas de opressão e controle mais amplos, e isso só é possível ao direcionar o foco não apenas para aqueles impactados negativamente por sistemas digitais e inteligência artificial, mas também para aqueles que detêm poder e se beneficiam dos algoritmos [Barabas et al. 2020]. O melhor caminho para a computação e engenharia de software, talvez, não seja a construção de métodos formais de avaliação de justiça, mas sim a condução de um processo no qual o usuário seja também um indivíduo inserido em um contexto social e político. Nesse processo, os requisitos devem ir além dos critérios funcionais e não funcionais, incorporando também compromissos políticos. Os testes de justiça, por sua vez, não devem se limitar à avaliação da validade de uma decisão, mas devem considerar quem realmente se beneficia dela, mesmo quando esta é considerada justa.

5. Considerações finais

Nesse trabalho, propomos uma discussão sobre alguns limites das abordagens propostas para a criação e validação de sistemas algorítmicos e de inteligência artificial justos. O campo de estudo sobre justiça algorítmica é recente e, apesar da revisão de literatura realizada indicar uma tendência às abordagens instrumentalistas e formais, não existe consenso na maneira a qual a ideia de justiça algorítmica é operacionalizada. Este torna-se, então, um momento oportuno para investigar os problemas e as contradições emergentes da justiça algorítmica.

Para isso, é fundamental não se ater unicamente ao questionamento sobre como construir algoritmos com tomadas de decisões justas. Compreender que o campo de estudo em questão não existe de forma fixa que precede a própria investigação conduzida por cientistas da computação e engenheiros de software abre possibilidades para outras perguntas: De que maneiras os algoritmos se entrelaçam com estruturas sociais e contribuem para a produção de desigualdades? Quem ganha com a persistência de sistemas algorítmicos injustos e discriminatórios? Como a ciência da computação e a engenharia de software podem trazer esse questionamento às suas práticas? E, sobretudo, o que se ganha e o que se perde ao analisar os impactos dos algoritmos a partir da perspectiva predominante na ciência da computação e engenharia de software?

Referências

- Abdu, A. A., Pasquetto, I. V., and Jacobs, A. Z. (2023). An empirical analysis of racial categories in the algorithmic fairness literature. In *2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 1324–1333. ACM.
- Adebayo, J. A. (2016). Fairml : Toolbox for diagnosing bias in predictive modeling.
- Alencar, I. (2023). Com mais de mil prisões na BA, sistema de reconhecimento facial é criticado por racismo algorítmico; inocente ficou preso por 26 dias. Accessed: 2024-10-27.
- Angwin, J., Larson, J., Mattu, S., and Kirchner, L. (2016). Machine bias: There’s software used across the country to predict future criminals. and it’s biased against blacks. Accessed: 2024-10-27.
- Barabas, C., Doyle, C., Rubinovitz, J., and Dinakar, K. (2020). Studying up: reorienting the study of algorithmic fairness around issues of power. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* ’20, page 167–176, New York, NY, USA. Association for Computing Machinery.
- Calmon, F. P., Wei, D., Ramamurthy, K. N., and Varshney, K. R. (2017). Optimized data pre-processing for discrimination prevention.
- Caton, S. and Haas, C. (2024). Fairness in machine learning: A survey. *ACM Comput. Surv.*, 56(7).
- Cavalcante, L. T. C. and de Oliveira, A. A. S. (2020). Métodos de revisão bibliográfica nos estudos científicos. *Psicologia em Revista*, 26:83 – 102.
- Chen, Z., Zhang, J. M., Hort, M., Harman, M., and Sarro, F. (2024). Fairness testing: A comprehensive survey and analysis of trends. *ACM Trans. Softw. Eng. Methodol.*, 33(5).
- Danilo Vieira and Mariana Cardoso (2024). Motorista de aplicativo alvo de tiros que mataram turista baiana diz que não houve ordem de parada por traficantes.
- de Souza Santos, R., Fronchetti, F., Freire, S., and Spinola, R. (2025). Software fairness debt.
- Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. (2011). Fairness through awareness.
- Flick, U. (2014). *The SAGE Handbook of Qualitative Data Analysis*. SAGE Publications Ltd.
- Greene, D., Hoffmann, A. L., and Stark, L. (2019). Better, nicer, clearer, fairer: A critical assessment of the movement for ethical artificial intelligence and machine learning. In *Hawaii International Conference on System Sciences*.
- Hardt, M., Price, E., and Srebro, N. (2016). Equality of opportunity in supervised learning.
- Hoffmann, A. L. (2019). Where fairness fails: data, algorithms, and the limits of antidiscrimination discourse. *Information, Communication & Society*, 22(7):900–915.
- Kamiran, F. and Calders, T. Data preprocessing techniques for classification without discrimination. 33(1):1–33.

- Keele, S. et al. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical report, Technical report, ver. 2.3 ebse technical report. ebse.
- Kitchenham, B., Pretorius, R., Budgen, D., Pearl Brereton, O., Turner, M., Niazi, M., and Linkman, S. (2010). Systematic literature reviews in software engineering – a tertiary study. 52(8):792–805.
- Malazita, J. W. and Resetar, K. (2019). Infrastructures of abstraction: how computer science education produces anti-political subjects. 30(4):300–312.
- Mendonca, R. F., Almeida, V., and Filgueiras, F. (2023). *Algorithmic Institutionalism: The Changing Rules of Social and Political Life*. Oxford University Press.
- Palumbo, G., Carneiro, D., and Alves, V. (2024). Objective metrics for ethical AI: a systematic literature review.
- Pham, N., Pham-Ngoc, H., and Nguyen-Duc, A. (2023). Fairness requirement in AI engineering – a review on current research and future directions. volume 62, pages 3–13.
- Rabonato, R. T. and Berton, L. (2024). A systematic review of fairness in machine learning.
- Roberta de Souza (2024). Motorista de aplicativo morre baleado após entrar por engano em comunidade de campo grande; passageiros também foram alvejados.
- Schwöbel, P. and Remmers, P. (2022). The long arc of fairness: Formalisations and ethical discourse. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 2179–2188. ACM.
- Soremekun, E., Papadakis, M., Cordy, M., and Traon, Y. L. (2022). Software fairness: An analysis and survey.
- Stern, P. N. (1997). Qualitative meta-analysis. *Completing a Qualitative Project: Details and Dialogue*, page 311.
- Trigo, A., Stein, N., and Belfo, F. P. (2024). Strategies to improve fairness in artificial intelligence:a systematic literature review. *Education for Information*, 40(3):323–346.
- Udeshi, S., Arora, P., and Chattopadhyay, S. (2018). Automated directed fairness testing. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, ASE ’18*, page 98–108.
- Verma, S. and Rubin, J. (2018). Fairness definitions explained. In *Proceedings of the International Workshop on Software Fairness, FairWare ’18*, page 1–7, New York, NY, USA. Association for Computing Machinery.
- West, S. M. (2020). Redistribution and rekognition: A feminist critique of algorithmic fairness. 6(2).
- Wu, Z. and He, J. (2022). Fairness-aware model-agnostic positive and unlabeled learning. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, FAccT ’22*, page 1698–1708, New York, NY, USA. Association for Computing Machinery.
- Zhang, B. H., Lemoine, B., and Mitchell, M. (2018). Mitigating unwanted biases with adversarial learning.