REDIRECTIONS

Editor: **Tim Menzies**
North Carolina State University
tim@menzies.us

Think Your Artificial Intelligence Software Is Fair? Think Again

Rachel K.E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang

From the Editor

Artificial intelligence software is still software. When software goes wrong, engineers must step in to fix the problem. In this article, researchers from IBM discuss how engineers can understand and fix issues related to discrimination resulting from the application of machine-learning software. —*Tim Menzies*

TODAY, MACHINE-LEARNING software is used to help make decisions that affect people's lives. Some people believe that the application of such software results in fairer decisions because, unlike humans, machine-learning software generates models that are not biased. Think again. Machine-learning software is also biased, sometimes in similar ways to humans, often in different ways. While fair model-assisted decision making involves more than the application of unbiased models—consideration of application context, specifics of the decisions being made, resolution of conflicting stakeholder viewpoints, and so forth—mitigating bias from machine-learning software is important and possible but difficult and too often ignored.

Algorithmic decision making has entered many high-stakes domains, such as finance, hiring, admissions, criminal justice, and social welfare. And in some cases, models generated from machine-learning software are found to make better decisions than humans can alone.[1,2] There are many examples to the contrary, however, where the models made by machine-learning software have been found to exacerbate bias and make arguably unfair decisions. Noteworthy examples include the following.

- Deployed sentiment-analysis models that determine the degree to which sentences express a negative or positive sentiment have been shown to be unacceptably biased,[3] giving negative scores to sentences such as "I am a Jew," and "I am homosexual."
- Deployed photo-tagging models have assigned animal-category labels to dark-skinned people.[4]
- Recidivism-assessment models used by the criminal justice system to inform decisions about who can be set free have been found to be more likely to falsely label black defendants as future criminals at almost twice the rate as white defendants.[5]
- Deployed facial-recognition software used to predict characteristics, such as gender, age, and mood, has been found to have a

Digital Object Identifier 10.1109/MS.2019.2908514
Date of publication: 18 June 2019

much higher error rate for dark-skinned women compared to light-skinned men.[6]

- Predictive policing software used to deploy police to where they are most likely needed has been found to overestimate crime rates in certain areas without taking into account the possibility that more crime is observed there simply because more officers have been sent there in the past.[7]
- An effort to create a job-recruiting application to automate the search for top talent was abandoned after years of work because it showed bias against women.[8]

Books, such as Cathy O'Neil's *Weapons of Math Destruction*,[9] provide even more examples of unfair decisions being made by software and argue that machine-learning software generates models that are full of bias. Hence, this is one of the

TESTING AND FAIRNESS IN THE SOFTWARE ENGINEERING LITERATURE

Issues of fairness have been explored in many recent papers in the software engineering research literature. Angell et al.[S1] argue that issues of fairness are analogous to other measures of software quality. Brin and Meliou[S2] discuss how to efficiently generate test cases to check for discrimination, and Başak Aydemir and Dalpiaz[S3] review frameworks for helping stakeholders explore ethical issues. Udeshi's team[S4] shows how to generate discriminatory inputs for machine-learning software. Albarghouthi and Vinitsky[S5] explore whether fairness can be wired into annotations within a program, while Tramèr et al. propose different ways to measure discrimination.[S6]

References

S1. R. Angell, B. Johnson, Y. Brun, and A. Meliou, "Themis: Automatically testing software for discrimination," in *Proc. 2018 26th Association Computing Machinery Joint Meeting European Software Engineering Conf. Symp. Foundations Software Engineering*, pp. 871–875. doi: https://doi.org/10.1145/3236024.3264590.

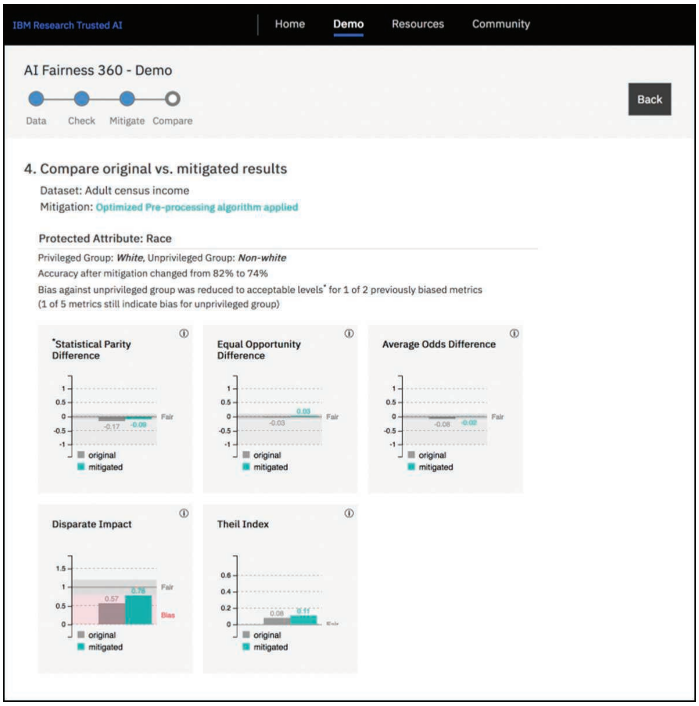
S2. Y. Brun and A. Meliou, "Software fairness," in *Proc. New Ideas and Emerging Results Track at 26th Association Computing Machinery Joint European Software Engineering Conf. Symp. Foundations Software Engineering*, 2018, pp. 754–759.

S3. F. Başak Aydemir and F. Dalpiaz, "A roadmap for ethics-aware software engineering," in *Proc. Int. Workshop Software Fairness*, 2018, pp. 15–21. doi: https://doi.org/10.1145/3194770.3194778.

S4. S. Udeshi, A. Pryanshu, and C. Sudipta, "Automated directed fairness testing." in *Proc. 2018 33rd Association Computer Machinery/IEEE Int. Conf. Automated Software Engineering*, pp. 98–108.

S5. A. Albarghouthi and S. Vinitsky, "Fairness-aware programming," in *Proc. Association Computer Machinery Conf. Fairness, Accountability, and Transparency*, 2019, pp. 211–219. doi: https://doi.org/10.1145/3287560.3287588.

S6. F. Tramèr et al., "FairTest: Discovering unwarranted associations in data-driven applications," in *Proc. 2017 IEEE European Symp. Security and Privacy (EuroS&P)*, pp. 401–416.



AI Fairness 360 - Demo

4. Compare original vs. mitigated results

Dataset: Adult census income

Mitigation: Optimized Pre-processing algorithm applied

Protected Attribute: Race

Privileged Group: *White*, Unprivileged Group: *Non-white*

Accuracy after mitigation changed from 82% to 74%

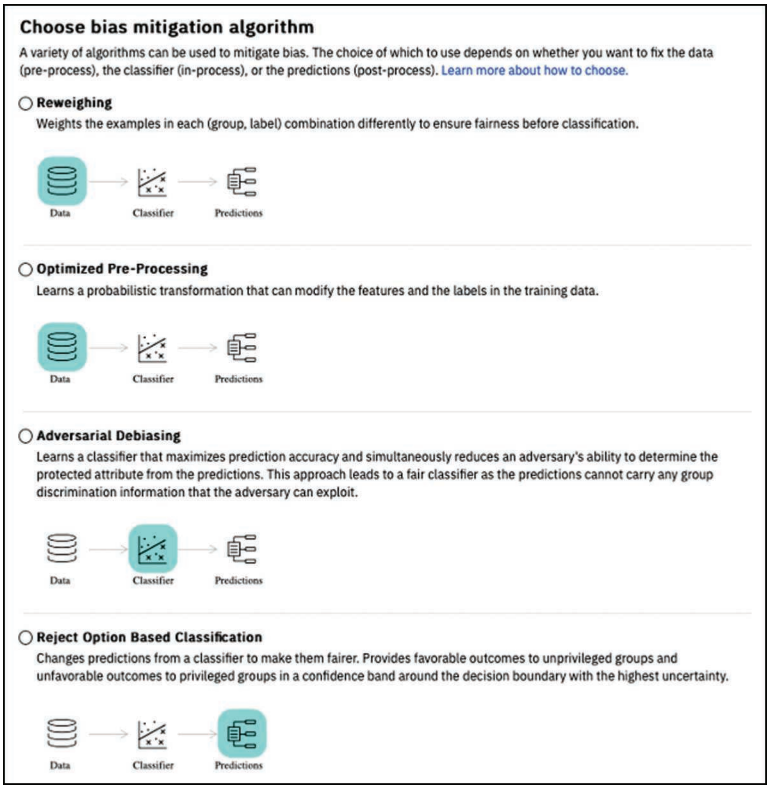
Bias against unprivileged group was reduced to acceptable levels* for 1 of 2 previously biased metrics (1 of 5 metrics still indicate bias for unprivileged group)

FIGURE 1. AIF360 toolkit resources. The website and interactive web experience can be found at http://aif360.mybluemix.net. The GitHub with the code and documented application programing interface can be found at https://github.com/ibm/aif360. Python project: https://pypi.org/project/aif360.

reasons that their application results in unfair decisions. The stakes for organizations and society are substantial. Clearly, there are potential benefits to the application of machine-learning software, such as increased productivity and reduction in human decision making bias. However, there are also potential downsides, such as significant public embarrassment and, most importantly, injustice.

Bias is such an issue because machine-learning software, by its very nature, is always a form of statistical discrimination. The discrimination becomes objectionable when it places certain groups or individuals at a systematic advantage and other groups or individuals at a systematic disadvantage. In certain situations, such as employment (hiring and firing), discrimination is not only objectionable but illegal.

Our vision is machine-learning software that can assist in recognition, repair, and explanation of biases. Achieving this vision is nontrivial. Recent years have seen an outpouring of research on fairness and bias in the models generated by machine-learning software. Narayanan[10] described at least 21 mathematical definitions of fairness in the literature. These are not just theoretical differences in how to measure fairness; different definitions produce entirely different outcomes. For example, ProPublica (an investigative news organization) and Northpointe (a company that creates case-management software for the judicial system) had a public debate on an important social-justice issue (recidivism prediction) that was fundamentally about what the right fairness metric is.[11–13] Also, researchers have shown that it is impossible to satisfy all definitions of fairness at the same time.[14] Further, in the software engineering (SE) literature, there is much interest in issues of fairness and testing (see "Testing and Fairness in the Software Engineering Literature"). Thus, although fairness research is a very active field, clarity on which bias metrics and bias-mitigation strategies are most appropriate for different contexts is yet to be achieved.



Choose bias mitigation algorithm

A variety of algorithms can be used to mitigate bias. The choice of which to use depends on whether you want to fix the data (pre-process), the classifier (in-process), or the predictions (post-process). Learn more about how to choose.

○ **Reweighing**
Weights the examples in each (group, label) combination differently to ensure fairness before classification.

○ **Optimized Pre-Processing**
Learns a probabilistic transformation that can modify the features and the labels in the training data.

○ **Adversarial Debiasing**
Learns a classifier that maximizes prediction accuracy and simultaneously reduces an adversary's ability to determine the protected attribute from the predictions. This approach leads to a fair classifier as the predictions cannot carry any group discrimination information that the adversary can exploit.

○ **Reject Option Based Classification**
Changes predictions from a classifier to make them fairer. Provides favorable outcomes to unprivileged groups and unfavorable outcomes to privileged groups in a confidence band around the decision boundary with the highest uncertainty.

FIGURE 2. Understanding bias-mitigation workflows in AIF360.

In addition to the multitude of fairness definitions, different bias-mitigation algorithms address different parts of the model lifecycle, and understanding how, when, and why to use each is challenging even for experts in algorithmic fairness. As a result, the general public, the fairness scientific community, and AI practitioners need guidance on how to proceed. To assist with the process of understanding and mitigating biases in models generated by machine-learning software, we have created AI Fairness 360 (AIF360); see Figure 1.

The original AIF360 Python package implemented techniques from eight published papers from the broader algorithm-fairness community. At the time of writing this article, two additional techniques had been added to the package, one added by IBM and the other by an external contributor to the project. AIF360 is designed as an end-to-end workflow with two goals—ease of use and extensibility: users should be able to easily go from raw data to a fair model, and researchers

ABOUT THE AUTHORS

RACHEL K.E. BELLAMY is a principal research staff member with IBM Research, Yorktown Heights, New York. Contact her at rachel@us.ibm.com.

KUNTAL DEY is a senior software engineer with IBM Research, New Delhi, India. Contact him at kuntadey@in.ibm.com.

MICHAEL HIND is a distinguished research staff member with IBM Research, Yorktown Heights, New York. Contact him at hindm@us.ibm.com.

SAMUEL C. HOFFMAN is a research software engineer with IBM Research, Yorktown Heights, New York. Contact him at shoffman@ibm.com.

STEPHANIE HOUDE is a designer with IBM Research, Yorktown Heights, New York. Contact her at Stephanie.Houde@ibm.com.

KALAPRIYA KANNAN is a research staff member with IBM Research, Bangalore, India. Contact her at kalapriya.kannan@in.ibm.com.

PRANAY LOHIA is a research software engineer with IBM Research, Bangalore, India. Contact him at plohia07@in.ibm.com.

SAMEEP MEHTA is a senior manager and is with IBM Research, Bangalore, India. Contact him at sameepmehta@in.ibm.com.

ALEKSANDRA MOJSILOVIC is an IBM fellow and head of the AI Foundation with IBM Research, Yorktown Heights, New York. Contact her at aleksand@us.ibm.com.

SEEMA NAGAR is an advisory researcher in artificial intelligence with IBM Research, Bangalore, India. Contact her at senagar3@in.ibm.com.

KARTHIKEYAN NATESAN RAMAMURTHY is a research staff member with IBM Research, Yorktown Heights, New York. Contact him at knatesa@us.ibm.com.

JOHN RICHARDS is a distinguished research staff member with IBM Research, Yorktown Heights, New York. Contact him at ajtr@us.ibm.com.

DIPTIKALYAN SAHA is a research staff member with IBM Research, Bangalore, India. Contact him at diptsaha@in.ibm.com.

PRASANNA SATTIGERI is a research staff member with IBM Research, Yorktown Heights, New York. Contact him at psattig@us.ibm.com.

MONINDER SINGH is a research staff member with IBM Research, Yorktown Heights, New York. Contact him at moninder@us.ibm.com.

KUSH R. VARSHNEY is a principal research staff member with IBM Research, Yorktown Heights, New York. Contact him at krvarshn@us.ibm.com.

YUNFENG ZHANG is a research staff member with IBM Research, Yorktown Heights, New York. Contact him at zhangyun@us.ibm.com.

should be able to contribute new functionality. A built-in testing infrastructure maintains code quality.

AIF360 is not just a Python package. It is also an interactive experience that provides guidance. The guidance explains that there are three main paths to the goal of making fairer predictions: fair preprocessing, fair in-processing, and fair postprocessing (Figure 2). Each corresponds to a category of bias-mitigation algorithms that we have implemented in AIF360. For example, preprocessing algorithms can be used when the original training data are available, in-processing algorithms can be used if the user can retrain the classifier, whereas postprocessing algorithms apply to existing classifiers without retraining. Users have the flexibility to try all categories of bias mitigation algorithms when they can touch all parts of the pipeline.

AIF360 comprises four classes: data set, metrics, explainer, and algorithms. The data set class and its subclasses handle all forms of data. Training data are used to instruct classifiers. Testing data are used to make predictions and compare metrics. Beside these standard aspects of a machine-learning pipeline, fairness applications also require associating protected attributes with each instance or record in the data. To maintain a common format, independent of what algorithm or metric is being applied, we chose to structure the data set class so that all of these relevant attributes—features, labels, protected attributes, and their respective identifiers (names describing each)—are present and accessible from each instance of the class. The metrics class and its subclasses compute various individual and group fairness metrics to check for bias in data sets and models. The explainer class is intended to be associated with the metrics class and provide descriptions of how fairness metrics are computed. The algorithms class implements bias-mitigation algorithms that can be applied at different points in the machine-learning pipeline.

There is a lot of work left to do to achieve unbiased AI. Fairness is a multifaceted, context-dependent social construct that defies simple definition. More work is needed to

- extend and apply the AIF360 toolkit to additional data sets and situations
- add other fairness measures
- add new applications, for example, how to determine fair pay for all workers regardless of gender or race
- extend the variety of explanations offered
- create guidance for practitioners on when a specific kind of explanation is most appropriate.

We invite you to offer your own approaches to fairness and bias checking, mitigation, and explanation to the tool kit. Your contributions would be most welcome!

References

1. I. Erel, L. Stern, T. Chenhao, and M. S. Weisbacj, “Could machine learning help companies select better board directors?” *Harvard Business Rev.*, Apr. 9, 2018. [Online]. Available: https://hbr.org/2018/04/research-could-machine-learning-help-companies-select-better-board-directors
2. S. W. Gates, V. G. Perry, and P. M. Zorn, “Automated underwriting in mortgage lending: Good news for the underserved?” *Housing Policy Debate*, vol. 13, pp. 369–392, 2002. doi: 10.1080/10511482.2002.9521447.
3. A. Thompson, “Google’s sentiment analyzer thinks being gay is bad,” Motherboard, Oct. 25, 2017. [Online]. Available: https://motherboard.vice.com/en_us/article/j5jmj8/google-artificial-intelligence-bias
4. A. Schupak, “Google apologizes for mis-tagging photos of African Americans,” CBS News, July 1, 2015. [Online]. Available: https://www.cbsnews.com/news/google-photos-labeled-pics-of-african-americans-as-gorillas/
5. J. Angwin, J. Larson, S. Mattu, and L. Kirchner, “Machine bias: There’s software used across the country to predict future criminals. And it’s biased against blacks,” ProPublica, May 23, 2016. [Online]. Available: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing
6. L. Hardesty, “Study finds gender and skin-type bias in commercial artificial-intelligence systems,” MIT News, Feb. 11, 2018. [Online]. Available: https://news.mit.edu/2018/study-finds-gender-skin-type-bias-artificial-intelligence-systems-0212
7. M. Reynolds, “Biased policing is made worse by errors in pre-crime algorithms,” *New Scientist*, Oct. 4, 2017. [Online]. Available: https://www.newscientist.com/article/mg23631464-300-biased-policing-is-made-worse-by-errors-in-pre-crime-algorithms/
8. J. Dastin, “Amazon scraps secret AI recruiting tool that showed bias against women,” Reuters, Oct. 9, 2018. [Online]. Available: https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-iduskcn1mk08g
9. C. O’Neil, *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*, New York: Crown, 2016.
10. A. Narayanan, “Translation tutorial: 21 fairness definitions and their politics,” presented at the Association Computing Machinery Conf. Fairness, Accountability, and Transparency, New York, Feb. 2018.
11. W. Dieterich, C. Mendoza, and T. Brennan, “COMPAS risk scales: Demonstrating accuracy equity and predictive parity,” Northpointe, Inc., Traverse City, MI, 2016. [Online]. Available: https://assets. documentcloud.org/documents/2998391/ProPublica-Commentary-Final-070616.pdf
12. J. Larson and J. Angwin, “Technical response to Northpointe,” ProPublica, July 29, 2016. [Online]. Available: https://www.propublica.org/article/technical-response-to-northpointe
13. J. Larson, S. Mattu, L. Kirchner, and J. Angwin, “How we analyzed the COMPAS recidivism algorithm,” ProPublica, New York, May 23, 2016. [Online]. Available: http://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm
14. J. Kleinberg, S. Mullainathan, and M. Raghavan, “Inherent trade-offs in the fair determination of risk scores,” in *Proc. Innovations Theoretical Computer Science*, 2017. doi: 10.4230/LIPIcs.ITCS.2017.43.