# AFOG workshop panel 4: From the black box society to the audit society — are algorithms auditable?

By Andrew Smart

Published August 13, 2018

Berkeley SCHOOL OF INFORMATION

Google

# INTRODUCTION

One of the topics discussed at the AFOG workshop was the role that algorithmic audits might play in ensuring algorithmic fairness and accountability. The question is how to develop rigorous approaches that can determine whether algorithms behave in undesirable ways and may also introduce a higher standard of accountability for companies and public institutions designing and programming algorithms. In the ideal case, algorithmic audits should give a voice to all stakeholders. Given the steady stream of cases of algorithmic unfairness both in the media and in academic research, and the recognition that algorithmic decisions will become more pervasive and more consequential, there is an urgency around holding algorithms accountable for their actions. In keeping with this trend, a commercial race seems to be emerging between nascent startups and small consultancies and large established companies offering algorithmic audits for bias and unfairness.

As Kroll (2015) points out, “In general, accountability fosters important social values, such as fairness, transparency, and due process, each of which is a deep subject addressed by an enormous literature.” The goal of having accountable algorithms is to make sure that the increasing use of algorithms in every domain furthers these social values. Algorithmic audits, like audits in more established industries, are techniques for verifying that technological and business practices are accountable to important social values. The ideal purpose of audits is to enable society to verify that its expectations of public and private organizations are met, and to offer a way to articulate what it means for organizations to behave in an accountable and responsible manner, and to provide evidence of performance to a standard (internal or external). Audits are also a technique for controlling risk in large organizations.

A recurrent theme from panelists on audits was how the culture of tech has traditionally favored a rapid product development cycle, whereas “auditing” requires slowing down to check that the concerns of many stakeholders are addressed, and that the standards against which one is being audited are met. Underlying this clash of cultures for algorithmic auditing for fairness are the problems of *what* and *whose* values get to decide what is fair? The challenge is to define

what the goals or standards are, and then how can companies and practitioners show outsiders that they are working toward or achieving those goals. Even if tech companies begin auditing their algorithms and development processes against fairness criteria, this is no guarantee that algorithms will become fair. The hope is that, at the least, they will be less unfair. There are also multiple sources of external unfairness in society and the question is to what extent do algorithmic systems reinforce existing social stratification, create new forms of stratification, and what algorithms can do to mitigate social injustice?

The following sections give an overview of what audits are, a brief history of audits and the concerns that motivate auditing in several areas: sociology, law, medicine, safety-critical industries, and finance. We also try to draw lessons for algorithmic auditing from these domains. Finally we examine future directions for research and how to move forward in a thoughtful way.

## WHAT IS AN AUDIT?

Audits are tools for interrogating complex processes to determine whether these processes are compliant with company policy, industry standards or regulations. While there are many similarities across fields in terms of what an audit is, there are key differences, too. Power (1997) describes the myriad types of audits, “In addition to financial audits, there are now environmental audits, value for money audits, management audits, forensic audits, data audits, intellectual property audits, medical audits, teaching audits, technology audits, stress audits, democracy audits and many others besides.” And we now add to that list, “algorithmic audits.” Audits may be considered what Floridi (2014) calls *metatechnologies,* or second- and third-order technologies that operate on and regulate other technologies; metatechnologies include the sociotechnical conditions in which the technology is embedded (e.g., rules, conventions, and laws).

For clinical trials in medicine for example, an audit is defined as “a systematic and independent examination of trial related activities and documents to determine whether the evaluated trial

related activities were conducted, and the data were recorded, analyzed and accurately reported according to the protocol, the sponsors standard operating procedures, Good Clinical Practice, and the applicable regulatory requirements." The IEEE defines an audit for software as, "an independent evaluation of conformance of software products and processes to applicable regulations, standards, guidelines, plans, specifications, and procedures." The idea of a *systematic* and *independent examination* or *evaluation* is a common ideal for audits in general. The audit is meant to check that a process and its result were done according to a value-based and agreed-upon set of rules or standards.

Ideally, audits are concerned with not only the output of a specific system, but also with the checks, controls, and quality of the system generating the output. A system with poor quality controls may produce good outputs by chance, but there may be a high risk of the system producing an error unless the controls are improved. The theory is that if an organization has sufficient control over its processes, whatever downstream or emergent property is desired--e.g., "airline safety", "pharmaceutical safety", "financial trust," and now "algorithmic fairness"--can be guaranteed (or at least the likelihood of a failure can be decreased) by an auditable (i.e., controllable) process. Auditing itself is its own industry with experts, literature, and conferences. And it's not that large tech platforms don't already audit many of their processes: financial auditing is a well-developed practice in the tech industry.

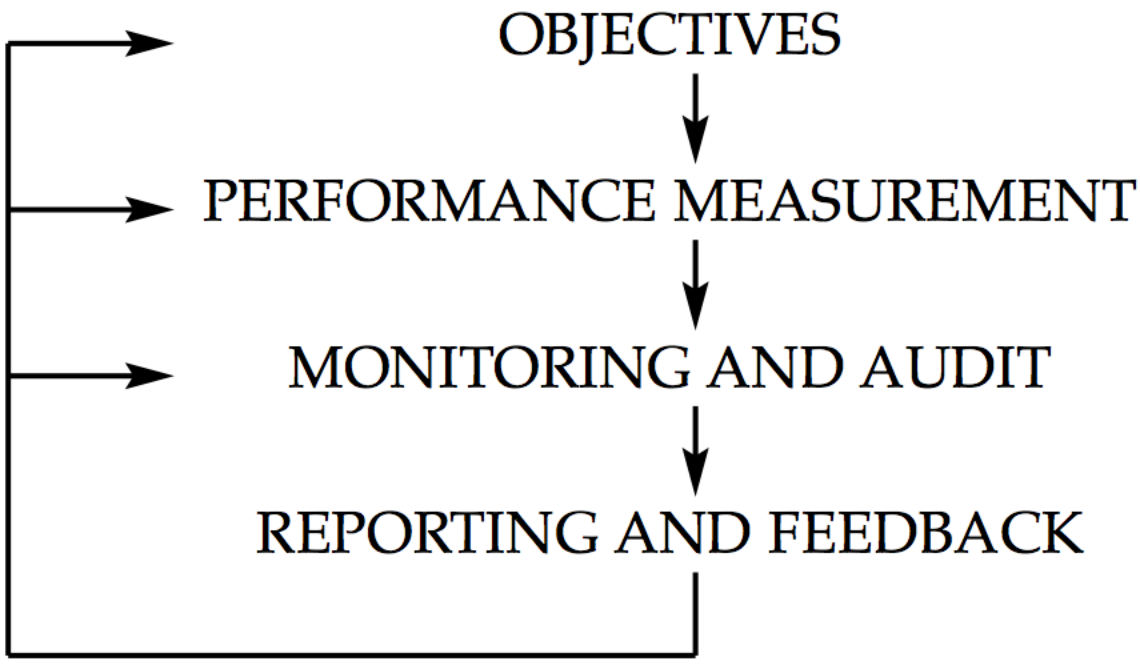


*Figure 1:* **A General Model of Self-Auditing and Control**

Figure 1 is taken from the ISO 9000 standard for quality assurance. The idea is to enable organizations to establish their procedures for ensuring quality, what Power (1999) calls "control of control." For machine learning, one might speak of "algorithms of algorithms."

Subjecting the machine learning technology development process to audits whether internal or external is not common practice. Are algorithms even the kinds of objects that can be audited? Burrell (2016) points out that algorithms and the organizational decision making that produces them are often deliberately opaque, and algorithms are protected as trade secrets and to prevent hostile actors from manipulating online services. However, these are justifications for resisting external audits, but are not practical reasons preventing audits. Then there is the inherent inscrutability of deep-learning models (Selbst and Barocas, 2018). Complete

transparency in terms of releasing code is has drawbacks and seems poorly suited to meeting fairness concerns. Transparency also has several serious disadvantages in terms of privacy and the risks of bad actors gaming systems. Using a technique known as “model inversion” it is already possible to infer private data from analyzing algorithms (Veale, 2018). Sandvig (2014) proposes five types of potential algorithmic audits which could answer questions about human values: 1) code audits; 2) noninvasive user audits; 3) scraping audits; 4) sock puppet audits; and 5) crowdsourced/collaborative audits. However as Deirdre Mulligan points out, in civil rights law items 2 & 4 from Sandvig’s list would be considered testing and distinct from auditing. “Testing” is a common practice when officials are looking for evidence of dicrimination. Kroll et al., (2017) propose a type of partial transparency using cryptographic techniques to evaluate if an algorithm meets certain independently-defined value criteria, such as fairness, without needing to explicitly check source code.

## WHAT SHOULD WE AUDIT?

It is therefore important to distinguish what exactly might be audited when we discuss auditing algorithms, and what standards - processes or outcomes - they should be audited against. Normatively, an external auditor should inspect a machine-learning product development process to ensure that proper safeguards are in place to protect human values such as fairness. As mentioned previously, this would require process-documentation and outputs from the models to check against predefined and agreed-upon metrics. These metrics would have to be designed so as to capture the values we care about.

Sandvig (2014) and Kroll (2017) propose technical and computational solutions to auditing algorithms. This is likely more feasible given the aforementioned opaque way in which algorithms are designed and the secrecy surrounding the development process at private companies. Internal auditing could also take the form of evaluating the product development process itself, as is done in other regulated industries. This would require that these companies pursue “auditability” in terms of design plans and documentation. There are open questions around the lack of standards around algorithm development. What sorts of artifacts or

documents could an auditor demand that might satisfy some predefined criteria? However, even if an auditor were to find something "adverse" during the audit, who is accountable? In the case of aviation, medicine, and finance, there are regulatory bodies with varying degrees of legal authority to impose fines, take away certifications, or even demand documentation from companies.

In computer science there is a distinction between "black box" auditing where the auditor only has access to the output of the system, and "white box" auditing where an auditor knows the internal workings of the program and/or the processes involved in the development of the system. However, neither black or white box auditing guarantees that the root cause of a behavior can be discovered. The difference between black box versus white box auditing is mirrored in the distinction between internal auditing and external auditing. Most companies in regulated industries like pharmaceuticals perform internal audits, and they are subject to external audits. Often, these companies conduct internal audits in order to prepare for external audits.

A typical machine learning development process might be as follows: 1) product concept 2) product design 3) data collection 4) data processing 5) model selection or design 6) model training 7) model evaluation 8) product testing, and, finally, 9) putting the trained model into production systems. Importantly, the joint statistical distribution of the training and test data are not the same as the underlying distribution of new data from the real world after the model is in production. In other words, the model used in production is using a certain, more or less "representative" distribution, but that is different from the "real world" distribution about which the model is making predictions.

At each stage of development, there may be several organizations or teams involved in decision making. The goal is usually to make sure a product launches and "lands" successfully according to success metrics set out for the product. These can be any number of key performance indicators (KPIs); for example, an increase in Daily Active Users (DAUs) or number

of downloads. Many attempts are now being made to include fairness metrics in these objectives.

In safety-critical industries and finance, the governance of organizational, technological and managerial processes are constructed to be transparent and auditable. It's also important to note that appropriate documentation for auditing purposes is also intended to improve an organization's control over its product development. Finally, the concept of "tracability" is crucial in these industries: the behavior of the final built system, its components, their origin, and all key decisions should be traceable back to the original design requirements and intended use or function.

Larger research questions for industry and academia include the following:

1) What can (and can't) we borrow from other industries and practices and apply to the unique issues surrounding audits of algorithms?
2) What would an algorithmic audit look like in practice? What would we have to adapt from other types of audits to suit machine-learning algorithms? What if audits themselves are opaque?
3) Is the algorithm the right target for an audit, or would it be wiser to look at the larger system in which the algorithm operates?
4) How can we avoid some of the pitfalls of creating a mere check-box activity or an "audit society" with layers of bureaucracy?

## A BRIEF HISTORY OF AUDITS

> "Wherefore in all great works are Clerks so much desired?
> Wherefore are Auditors so well fed…?
> Because that by number such things they finde,
> which else would farre excell mans minde."
>
> Robert Recorde (1540)

Concerns about transparency and accountability are nothing new of course. Health and safety, medicine, security, education, intellectual property, aviation, discrimination law and policy, and corporate finance each have long histories and interactions with audits. It is also interesting to consider that auditing is fundamentally driven by calculation and quantification, just as algorithms are. Yet there is a sense now that Big Data and predictive algorithms, which quantify on an unprecedented scale, have become unaccountable. It's useful to review the history of auditing in other domains to draw lessons that may inform algorithmic audits.

## Audit studies in law and sociology

Audit studies, most often conducted in law and sociology to detect housing or job discrimination, date from the WWII era. An audit study to detect discrimination is a type of field experiment that stages randomized encounters between auditors and decision makers (e.g., landlords). The auditors are as closely matched as possible in all features and characteristics except the one under investigation (usually race or gender). In the case of racial discrimination in housing, for example, the theory is that if the landlord offers a home to a white person over a "matched-in-all-other-aspects" black person, then the researcher can infer that racial discrimination motivated the decision. Thus, "race" here becomes an isolated treatment effect, similar to the drug/placebo given in a randomized controlled clinical trial. Kohler-Haussman argues, however, that "audit studies do not measure the objective isolated treatment effect of race and race alone because there is no such thing to measure." In other words, treating race as an isolatable trait ignores the thick ethical and sociological underpinnings of race as a social construct. How social categories like race and gender are constituted is an important consideration for algorithmic fairness and auditing algorithms *qua* decision makers, as well.

For example in the criminal justice domain, the much-discussed COMPAS algorithm for recidivism risk was audited by its designers and found to be "good", but when ProPublica used different metrics based around criminal defendants' point of view, the COMPAS algorithm was

clearly unfair. The question is what metrics should an organization choose to audit against? And who should have a say in the choice of metrics? Echoing a theme that emerged during earlier panel discussions on evaluating algorithmic fairness, what is meaningful to measure and how can we know what those measurements are? For algorithmic fairness audits, we must define our objectives and how to measure performance. Defining and measuring fairness is a notoriously difficult task in machine learning (Kleinberg, 2016).

## Finance audits

Financial auditing had to play catch up as the complexity and automation of many financial business practices became too unwieldy to manage manually, thus stakeholders in large companies, and government regulators desired a way to hold companies accountable. Concerns among regulators and shareholders that the managers in large financial firms would squander profits from newly created financial instruments prompted the development finance audits.

Additionally, as financial transactions and markets became more automated, abstract, and opaque, threats to social and economic values were answered increasingly with audits. But financial auditing lagged behind the process of technology-enabled financialization of markets and firms. Similarly today, as algorithms invade more and more aspects of life, audits are proposed as an answer to perceived threats to important values like privacy, autonomy, and ethical behavior - but a framework to audit algorithms is lagging behind.

However, the limits of auditing have been seen most notably after the financial crash of 2008. Financial auditing was widely criticized in the wake of the 2008 financial crash as most firms that suddenly failed had received positive audit results immediately prior to public declaration of financial difficulties (Sikka, 2009: 869 - cited in Styhre, 2015: 147).

## Safety engineering

The fields of fault tolerance and safety engineering have developed a rich set of tools and techniques for analyzing high-assurance and fault-tolerant systems and their risks. These include analyses of systems like avionics on commercial aircraft and submarines. But it is generally recognized that audits are no guarantee of safety. Indeed, audits can provide a false sense of security and an organization can slowly drift toward unsafe decisions. If audits become a mindless check-box activity, then they might be used to claim an organization is doing something it is actually not. And once safety standards are relaxed, they typically don't snap back until there is a catastrophe.

As mentioned, the use of audits as a means of controlling risk in other industries reveals a fundamental cultural difference between safety-critical and the tech industries: risk-aversion, regulation and safety-first thinking using the precautionary principle for the former versus "move fast and break things" for the latter (or as one panelist put it, "slow down and do a good job" versus "permissionless innovation"). In contrast to the "ship it and fix it later" ethos that has defined the tech industry, safety engineering requires that the developer define what must be avoided (e.g., airplane crashes, patient death) and engineer backwards from there.

## Medicine and pharmaceutical

Internal and external Quality Assurance audits are a daily occurrence in the pharmaceutical industry, and the documentation and audit trails are as important as the drug products themselves. But large pharmaceutical companies have become enormous bureaucracies with thousands of people devoted to documenting and auditing the drug development process (and then documenting and auditing the auditing process). Crucially, the Food and Drug Administration (FDA) and the European Medicines Agency (EMA) are regulatory agencies with the legal authority to close down a company that routinely fails audits or cannot produce reliable documentation.

In medicine, the stages of product development are strictly defined. In fact, for medical devices, federal law (specifically, Code of Federal Regulations Title 21) mandates that medical-device makers establish and maintain what are called "design control" procedures in order to ensure

that design requirements are met. In other words, it is a legal requirement that medical devices be "auditable."

Medical-device makers must maintain procedures to ensure that these design requirements meet the "intended use" of the device. The intended use of a "device" (or, increasingly in medicine, an algorithm - see Price 2017 for more) determines the level of design control required: i.e., a tongue depressor (a simple piece of wood) is the lowest class of risk (Class I), while a deep brain implant would be the highest (Class III). The intended use of a tongue depressor could be "to displace the tongue to facilitate examination of the surrounding organs and tissues." The intended use is what differentiates a tongue depressor from a popsicle stick. This may be important when thinking about an algorithm that can be used to recommend movies or to identify tumors; depending on its intended use, the same algorithm might have drastically different risk profiles.

In legal regimes like this one covering medical devices, some software designers may get out of those documentation requirements altogether. For example, the 21st Century Cures Act, signed into law in late 2016, amended what gets classified as a "medical device" in the first place. So whether a certain "software function" (including clinical decision support software, which is one of the hot areas for machine learning in medicine) is regulated by the FDA as a "device" turns on statutory exclusions of the Food, Drug, and Cosmetic Act (FDCA) that were added by the Cures Act (see statutory language here and the FDA's draft guidance here and here). Within the context of decision support software, the FDA's initial interpretations of the amendments have elicited strong reactions from industry stakeholders, illustrating the complexities and contested nature of such a regulatory regime in the context of ML/AI.

In any case, for products classified as medical devices, at every stage of the development process, device makers must document the *design input, design output, design review, design verification, design validation, design transfer,* and *design changes.* All of this is kept in a Design History File (DHF), which must be an accurate representation of the product and its development process. Included in the DHF is an extensive risk assessment and hazard

analysis, which must be continuously updated if any new risks are discovered. When the FDA audits a pharmaceutical or device company, it will inspect the design history file. Finally, companies are required to proactively maintain what is called "post-market surveillance" for any issues that may arise with safety of a medical device.

### Security audits

Finally, in the computer security domain a successful ecosystem has evolved between tech companies and individual researchers finding security vulnerabilities. There are firms that do security audits that are distinct from internal or external vulnerability red teams. The latter often participate in what are called "bug bounties" where hackers are paid by websites to find and report bugs or security vulnerabilities. Elezari Bar On (2018) suggests modeling algorithmic auditing on the "white-hat" hacker bug bounty model adopted by the security industry, "we need a market that will facilitate a scalable, crowd-based system of auditing to uncover 'bias' and 'deceptive' bugs that will attract and galvanize a new class of white-hat hackers: algorithmic auditors. They are the immune system for the age of algorithmic decision-making." Within security research, some tools and practices that are used to test algorithms for bias may raise questions under the Computer Fraud and Abuse Act (CFAA) (Berkeley Center for Law & Technology Workshop Report, 2015).

## THE LIMITS OF AUDITS

An assumption underlying the justification for auditing is the idea that auditing simply produces objective facts. This is epistemologically problematic in itself, but also when combined with any potential financial interests among the auditors. Audits are not based on an Archimedian fixed point, but rather, like machine-learning models, they are based on the views and values of humans and organizations with particular interests. As discussed above auditing does not guarantee safety and can miss systemic risks, as the example of financial crisis spectacularly demonstrated.

Indeed as Kroll (2017) argues, even for highly technical evaluation of computer code, audits are limited in their ability to attribute cause to changes in system behavior, or explain why a particular change inputs had a meaningful impact on output.

From a social perspective, Power (2000) argues that it is "important to understand the growth and circulation of an idea of audit, a growth in which accountants have been powerful agents in selling their auditing capabilities but which cannot solely be explained in this way." Who are the agents selling their algorithmic auditing capabilities today? As mentioned, both large companies and startups are beginning to offer algorithmic audits as a service. Who are the algorithmic accountants? And who should hold these auditors accountable?

For a financial firm, for example, the auditing process is known and internal financial processes are designed to be "auditable." Not only is a deep neural network not designed to be auditable, it is difficult to see how its millions of weights and nonlinear functions could be auditible. Power (1996) says: "Making things auditable is a constant and precarious project of a system of knowledge which must reproduce itself and sustain its institutional role from a diverse assemblage of routines, practices and economics constraints." And even with access to "the guts of a system (the code, the architecture)", the policies and procedures that govern its development, and information about the use-environments and contexts, the values implications of design decisions may remain obscured (Mulligen & Bamberger, 2018). Ultimately, Power has argued audits risk becoming "performative rituals of verification," which might temporarily assuage our fears about inscrutable and uncontrollable processes or algorithms, but they are hardly cold, objective, independent producers of "facts". Therefore how we define what is to be audited becomes extremely important. Related to the fact that audit results too must be interpreted, Karl Popper pointed out, "Every observation and, to an even higher degree, every observation statement, is itself already an *interpretation in the light of our theories."*

Styhre (2015) describes the inherent embeddedness of audits and auditors within the very systems that should be audited, "The audit per se and its procedures and routines are never

isolated from the practices of the professionals conducting the work, and therefore the credibility of the professionals remains a key issue...the distinction between inside and outside, externality and interiority being of key importance for the legitimacy of the audit work is porous and fluid." There is no objective gaze from the audit. Financial auditing, for example, is fundamentally an *inferential* process which has to draw conclusions from a limited set of documents, budgets, oral testimony or direct observation (Power,2000). What might a similar sort of inferential auditing process look like for algorithms? Machine learning is also *inferential* and draws conclusions from limited datasets.

## RECOMMENDATIONS

As calls for the regulation of tech grow louder and more pointed, there is an opportunity for a closer collaboration among stakeholders in our increasingly information-based and algorithmic society. Well-thought through and voluntary algorithmic auditing practices could be used to demonstrate the ethical foresight necessary to decrease the probability of what Mittelstadt (2015) terms "regulatory whiplash", where "overly restrictive measures (especially legislation and policies) are proposed in reaction to perceived harms, which overreact in order to re-establish the primacy of threatened values."

(1) **An ethical algorithmic audit could address questions of fairness and accountability at each stage of product development** outlined above. This would require documentation to be generated making the product development process "auditable" such as audit trails and audit logs.The question of interiority versus exteriority is again important to consider, as external auditors in safety-critical industries are given access to internal documentation, design plans, and risk assessments.

(2) **Further research on what would be required to make algorithms and their development "auditable".** As discussed, neither algorithms nor the development process within most companies are designed to be auditable. Audit trails or audit logs are not generated

in standardized formats, nor are they used consistently for purposes such as determining fairness.

(3) There is an opportunity to create a **more meaningful dialogue among a wider array of stakeholders** in algorithmic accountablity, e.g., vulnerable communities impacted by technology, nonprofits on the frontlines of fighting discrimination and other forms of injustice, legislators and regulators, politicians, and the tech industry as a whole. Conferences on algorithmic accountability such as FAT* are a good start, but are still not well-known outside highly specialized academic disciplines. A broader public-oriented effort should bring all stakeholders to the table.

(4) **Do not make algorithmic audits as opaque as the algorithms they're meant to audit.** In the other industries discussed the process of auditing has become a professionalized endeavor, with highly specialized auditors who produce lengthy audit reports. The validity of the audit rests largely on the credibility of those professionals conducting the audit. Thus there is a risk for algorithmic auditing that, rather than revealing issues and improving accountability, algorithmic audit reports become interpretable to only a small group of highly trained experts - which is exactly one of the problems currently with algorithms that algorithmic audits are meant to address.

(5) In view of the complexity of both the algorithms themselves and the processes through which they are built (hundreds of engineers working across geographies), further research **should develop hybrid auditing techniques** that allow both internal and external stakeholders to trust their values are protected. These techniques should evaluate both the organizational processes and work practices, as well as the technical functioning of the systems. This could include everything from data generation and collection and model selection, but also quality assurance checks as the project advances.

(6) What would a standard for **good algorithmic development practice look like?** In accounting and finance, medicine, and other industries there are clear standards such as Generally Accepted Accounting Practices (GAAP) and Good Clinical Practice (GCP) for clinical

trials or Good Manufacturing Practice (GMP) for drug makers. Internal and external auditors then use these guidelines to find violations or to make recommendations to the organizations being audited.

## ACKNOWLEDGMENTS

We would like to thank the panelists on Panel 4: Auditing algorithms (from within and from without): Chuck Howell (MITRE), Danie Theron (Google), Michael Tschantz (International Computer Science Institute) and the moderator Allison Woodruff (Google). This blog is based on discussions from the panel, and on other input from many scholars and experts from industry who attended the workshop. We would also like to thank Daniel Kluttz, Jenna Burrell, Allison Woodruff, Deirdre Mulligan and Jen Gennai for helpful feedback and suggestions on prior drafts of this report. Finally, we acknowledge the intellectual support and organization from the AFOG group at UC Berkeley.

## ABOUT AFOG

The Algorithmic Fairness & Opacity Working Group (AFOG) is made up of UC Berkeley faculty, postdocs, and graduate students at UC Berkeley. It is housed at Berkeley's School of Information. AFOG is co-directed by Professors Jenna Burrell and Deirdre Mulligan and is funded by a research gift from Google Trust & Safety to support cross-disciplinary academic research and conversations between industry and academia to explore and address issues related to fairness and opacity in algorithms.

For more information visit: https://afog.berkeley.edu