When Trusted Black Boxes Don't Agree: Incentivizing Iterative Improvement and Accountability in Critical Software Systems

Jeanna Neefe Matthews
Graham Northup
Isabella Grasso
Stephen Lorenz
Marzieh Babaeianjelodar
Hunter Bashaw
Sumona Mondal
jnm,northug,grassoi,lorenzsj,babaeim,bashawhm,smondal@clarkson.edu
Clarkson University

Abigail Matthews
avmatthews@cs.wisc.edu
University of Wisconsin Madison

Mariama Njie
mnjie1@gaels.iona.edu
Iona College

Jessica Goldthwaite
JWGoldthwaite@legal-aid.org
The Legal Aid Society

ABSTRACT

Software increasingly plays a key role in regulated areas like housing, hiring, and credit, as well as major public functions such as criminal justice and elections. It is easy for there to be unintended defects with a large impact on the lives of individuals and society as a whole. Preventing, finding, and fixing software defects is a key focus of both industrial software development efforts as well as academic research in software engineering. In this paper, we discuss flaws in the larger socio-technical decision-making processes in which critical black-box software systems are developed, deployed, and trusted. We use criminal justice software, specifically probabilistic genotyping (PG) software, as a concrete example. We describe how PG software systems, designed to do the same job, produce different results. We highlight the under-appreciated impact of changes in key parameters and the disparate impact that one such parameter can have on different racial/ethnic groups. We propose concrete changes to the socio-technical decision-making processes surrounding the use of PG software that could be used to incentivize iterative improvements in the accuracy, fairness, reliability, and accountability of these systems.

CCS CONCEPTS

• **Software and its engineering** → **Software testing and debugging**; • **Social and professional topics** → *Governmental regulations.*

KEYWORDS

algorithmic accountability, criminal justice software, probabilistic genotyping, software verification, disparate impact

ACM Reference Format:
Jeanna Neefe Matthews, Graham Northup, Isabella Grasso, Stephen Lorenz, Marzieh Babaeianjelodar, Hunter Bashaw, Sumona Mondal, Abigail Matthews, Mariama Njie, and Jessica Goldthwaite. 2020. When Trusted Black Boxes Don't Agree: Incentivizing Iterative Improvement and Accountability in Critical Software Systems. In *Proceedings of the 2020 AAAI/ACM Conference on AI, Ethics, and Society (AIES '20), February 7–8, 2020, New York, NY, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3375627.3375807

1 INTRODUCTION

Software is increasingly used to direct and manage critical aspects of all of our lives from how we get our news, to how we find a spouse, to how we navigate the streets of our cities. Beyond personal decisions, software plays a key role in regulated areas like housing, hiring, and credit, as well as major public functions such as criminal justice and elections. Flaws in complex software systems are expected, but removing them can be difficult. We are accustomed to market forces incentivizing the costly process of debugging and iterative improvement. Unfortunately, in some of the most critical areas of software deployment, market forces may be utterly insufficient.

Consider the case of criminal justice software and in particular, probabilistic genotyping software that matches DNA evidence found at crime scenes to potential suspects. The case of the Forensic Statistical Tool (FST) developed by the Office of the Chief Medical Examiner (OCME) in New York City illustrates that developers may be tempted to avoid costly debugging by claiming intellectual property protection in order to keep knowledge of known problems away from defendants, defense teams, prosecution teams, experts, the scientific community, government oversight, and the public [8]. Is this an isolated incident or the symptom of a larger problem impacting critical software in criminal justice and many other areas of our lives?

AIES '20, February 7–8, 2020, New York, NY, USA

ACM ISBN 978-1-4503-7110-0/20/02…$15.00
https://doi.org/10.1145/3375627.3375807

In this paper, we discuss flaws in the larger socio-technical systems in which critical software is approved, chosen, deployed, and trusted. We use PG software as one concrete example, but we draw parallels to software in other areas. In addition, we propose concrete changes to these larger socio-technical systems that could and should be used to truly incentivize iterative improvements in critical software systems. As software is increasingly used in high-stakes decisions about the lives of individuals, it is essential that we question what will make that software responsive to the needs of individuals, to society, and to the law, not just the interests of those making the decisions or developing the software.

2 PROBABILISTIC GENOTYPING SOFTWARE AND EXISTING INCENTIVES FOR ITERATIVE IMPROVEMENT

When DNA evidence is found at a crime scene, it can be compared to the DNA of possible suspects. If there is a high quality evidence sample from one contributor, forensic analysts can often make the comparison manually, by visually comparing electropherograms generated from both the evidence sample and from a suspect's DNA. However, there are many things that can complicate this comparison. In standard DNA testing, Polymerase chain reaction (PCR) amplifies the DNA regions examined in order to allow for detection. However, when low amounts of DNA are being tested, stochastic effects interfere with the ability to accurately detect the DNA profile. Sometimes peak heights are skewed, alleles fail to be amplified (drop-out error) or artifacts are amplified (stutter) or contaminant alelles appear (drop-in error). This is a bit like errors that result from repeated photo-copying of an image. Errors are especially likely when the original DNA sample is small. Even the cells deposited by a single fingerprint are technically sufficient for testing, but results are more reliable with substantial quantities of DNA, such as that often found in blood or semen samples.

Another source of complication comes from multiple contributors to an evidence sample. The more individuals who contribute DNA to a sample, the more difficult it is to interpret, like it might become increasingly difficult to identify a single song being played when multiple songs are playing at once. Some contributors may have deposited more DNA than others, like some songs being played louder than others. Most labs validate PG software systems for only a small number of contributors (e.g. 1-4) and the person running the software must say how many contributors there are to a given sample. The setting of that parameter can have a substantial impact on the result, but in real case work, it is often impossible to know how many contributors really contributed to an evidence sample.

Consider cases of "Touch DNA" or DNA from fingerprints left on objects, for example a gun found at an outdoor crime scene. There may be little DNA present and what is present may be from an unknown number of contributors, perhaps vastly exceeding the number of contributors the software has been shown to reliably handle. The evidence sample may also have been degraded by being outside or contaminated by DNA that has been transferred to the scene without individuals actually being present. All of these possible sources of error add up and typically make it impossible for human analysts either to interpret the data manually or to double check software results. PG systems use advanced mathematics to make an informed statistical guess (probabilistic) about the source of DNA (genotyping) in a sample.

PG software systems have a complex set of parameters such as the number of contributors, estimates of the impact of drop-in errors and drop-out errors, the definition of a "random" individuals as specified by population frequency data, a parameter theta that reflects the degree of genetic diversity in a population, and many more. These parameters can have large impacts on the answer generated by a PG system and the uncertainty introduced when a forensic analyst selects a certain configuration is not always highlighted when presenting the data in court.

The key question we are asking in this paper is do sufficient incentives exist for flaws in the development and use of these complex systems to be identified and fixed? As citizens of the modern world, we are accustomed to bugs in complex software. We may, for example, experience a bug on our cell phone and find that the same odd problem is only appearing for us, not our friends or family. We may or may not report this bug ourselves, but we generally trust that bugs will be found and fixed eventually, that market forces will deliver incremental improvements. However, the same is not true for many critical black box systems, like PG software.

In the case of PG software, what happens when someone reports a problem? If a defendant reports that they have been incorrectly identified as a contributor to an evidence sample, will a bug report be investigated or will it simply be assumed that they are guilty? What if errors occur more often for some groups of people that others? Will forensic analysts simply trust the output of software if they cannot manually replicate the results? What if there is a serious bug in the system as there was in the case of the FST [8]? In that case, the developers knew clearly there was a problem that was causing their system to report impossible answers. OCME "fixed" the defect by dropping data that triggered the flaw even if that data might have been important to the defense or prosecution. They notified no one when data was dropped in a particular case and also aggressively resisted expert witness review that could have exposed the problem for 5 years while using the output of the system as evidence in over 1000 serious criminal cases [6].

What about market forces? FST was developed by a crime lab for in-house use. Would the incentive structure be different for a commercial product? Commercial developers might have more ability and incentive to improve their products, but they respond most directly to the interests of those purchasing their software. In criminal justice software and in many other examples of black-box decision-making software in areas like hiring or credit, the interests of those purchasing the software to make decisions can be very different than the interests of those being decided about. In criminal justice, for example, purchasers may want the software to help them interpret more DNA samples or close more cases, but that is different than being sure the samples are interpreted correctly or that the cases are closed correctly. Intellectual property rights were intended to benefit those who develop key advancements and share them with society. However, intellectual property rights can also be used to shield developers from identification of flaws they don't want to invest in fixing or to hide evidence that the system might be discriminating illegally [13].

3 WHEN TRUSTED BLACK-BOXES DON'T AGREE

Probabilistic genotyping (PG) software is designed to extend the forensic scientist's ability to analyze samples that are too complex for manual interpretation. These systems produce a likelihood ratio (LR) metric which is a complex statistical measure comparing the relative likelihoods of the data, given two competing hypotheses. Typically one is considered an incriminating 'prosecution hypothesis' and an exonerating 'defense hypothesis.'

The common formula for computing a likelihood ratio is $LR = Pr(E|Hp)/Pr(E|Hd)$ where E is evidence or observed data. The numerator represents one hypothesis (Hp) aligned with the prosecution position—usually that the suspect contributed DNA to the mixture, and the denominator represents the defense hypothesis (Hd) —that an unknown person left their DNA on the evidence. For samples containing DNA from multiple individuals, both Hp and Hd will include additional contributors, either assumed contributors whose genotypes are known or contributors whose identities are unknown. There are many known problems with the words used when presenting LRs in court, by journalists and even by trained forensic analysts. For example, an inclusionary LR is not a DNA match nor does an LR of X imply that it is X times more likely that the suspect is guilty. The probability of the evidence given the hypothesis that the person is guilty is not the same as the probability of guilt given the evidence. LR's report the first and juries are tasked with determining the second and they are not logically equivalent and rarely equal. A discussion of the misuses of LR as a metric are critical in how the output of these software systems are used in court but is beyond the scope of this paper [7].

An LR greater than 1 means there is more support for the prosecution hypothesis and is therefore considered inclusionary which is often inculpatory or suggestive of guilt. An LR less than 1 means there is more support for the defense hypothesis and is therefore exclusionary which is often exculpatory (suggestive of innocence).

Many probabilistic genotyping software systems have all been developed to perform this same task. Some are proprietary software packages produced by commercial companies, some are open source software packages produced by researchers, some are produced in-house by crime labs. The most common software packages used to introduce evidence in court varies by jurisdiction. Commercial software packages, STRmix and TrueAllele, are used widely throughout the United States. New York City used FST, a software package developed in-house by the Office of the Chief Medical Examiner from 2012 to 2017 until they largely replaced it with STRmix. Open-source systems LRMix and EuroForMix have been used widely for casework in Europe [**? ?**].

Although these systems are designed to answer the same forensic question, they can produce very different results and little effort has been made to standardize their behaviour or to provide automated testing interfaces that would allow them to be compared across a large number of test cases. Their results are not routinely compared in real casework. There have been academic publications comparing systems[4, 12], but more work is required to fully understand when, how and why these systems differ - between programs, within releases of the same program and with variations in parameters.

Evidence of problems has surfaced in real casework. For example, in a post-conviction matter, a fingernail sample which contained a mixture of the victim and the defendant was analyzed by two PG systems, TrueAllele and STRMix, which produced results that differed by many orders of magnitude. TrueAllele produced a exclusionary statistic in the trillions while STRmix produced an exclusionary statistic of 666 which fell within the laboratory's "uninformative" range. Later, a more complete DNA profile from the victim was generated and used in a subsequent STRmix analysis which generated a slightly larger exclusionary likelihood ratio of 1,980[1].

To explore differences between systems in a more systematic way, we added automated testing harnesses around three systems, FST, LRMix, and EuroForMix, in order to compare them on a set of test cases constructed by OCME for use in validating FST. These test cases consist of DNA mixtures created in a controlled laboratory setting and as a result, unlike with evidence samples in real casework, the true contributors and known non-contributors are clearly known. The OCME validation study and its data is described in more detail in Appendix A.

In Figure 1, we compare these three systems. The Y-axis shows the mean likelihood ratio (log scale) produced for both a set of true contributors to test samples (mean of approximately 1200 total LR results for true contributors) and a set of known non-contributors (mean of approximately 336,000 total LR results for known non-contributors). In some cases, individual systems especially FST produced an error or failed to produce a result. The mean graphed in Figure 1 includes only tests which produced a result. Results above 0 (log 1) are inclusionary and results below 0 are exclusionary. Notice that as expected the mean LR ratings for contributors are above 0 and the mean ratings for non-contributors are below 0. However, the differences among systems are meaningful.

We tested each system using its default configuration and also modified the configuration of LRMix and EuroForMix to make them as closely comparable to FST as we could ("As FST settings"). For FST itself, the "As FST" configuration is the same as the default configuration. For LRmix and EuroForMix we modified a number of the parameters used (e.g. parameters that estimate the rates of drop-in and drop-out errors) to more closely match FST. We will discuss the important impact of some of these parameters individually in more detail later in this paper.

It is notable that the drop-out parameters differ not only in their default settings, but also in the granularity at which they can be set and the way in which the parameters can be changed. For example, LRmix only allows one drop-out rate to be set for an entire case through the GUI. FST sets finer granularity per-locus drop-out rates that were determined in validation, but these can only be changed through an internal database. One result of this is that FST does not use a constant drop-out rate across all cases. When we run LRmix and EuroForMix in "As FST" mode, we attempt to vary the drop-out rates per case as FST does. This illustrates some of the fundamental problem due to the lack of standards across PG software. Even though they are designed to do the same job, they are not designed to be easily comparable. Establishing such standards is one key recommendation arising from our work.

We performed an Analysis of Variance (ANOVA) test on all configurations shown in Figure 1. The null hypothesis of an ANOVA is that the means of each data set are equal, and a p-value below 0.01

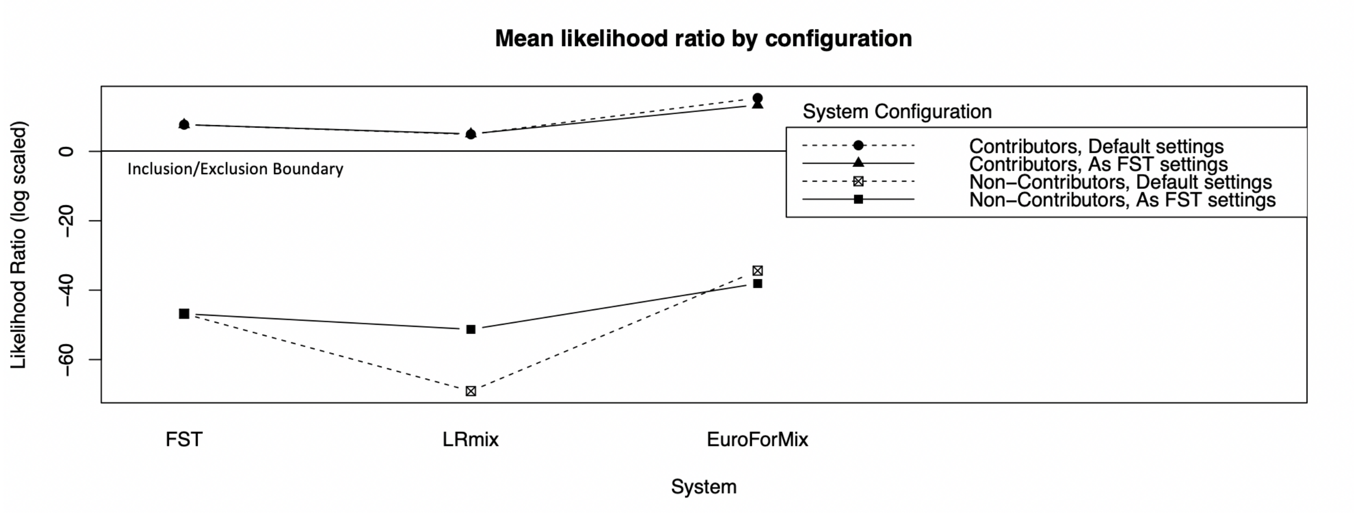

Figure 1: Y-axis shows the mean likelihood ratio (log scale) produced for both true contributors to test samples and for a set of known non-contributors. Results are shown for three PG systems: FST, LRmix and EuroForMix. We tested each system using its default configuration. We also modified the configuration of LRMix and EuroForMix to make them more closely comparable to FST. Results above 0 (log 1) are inclusionary and results below 0 are exclusionary.

infers a statistically significant difference. In all cases for our data, the p-values were less than 0.001, inferring statistically significant differences. When comparing the default settings for all systems, the F statistic, which gives a measure of how different the means are, was more than 8,000. When we tried to make the systems as comparable as possible (the "as FST" settings rather than the defaults), the F statistic was less than 100. Since the F statistic is so large, the discrepancies in the results between the systems are not due to variation within each system, but a fundamental difference between the systems. Thus, we were able to make the systems more directly comparable (lower F Statistic), but the differences that remain are still statistically significant (p-value less than 0.001). Since a significant ANOVA result only tells us that we can reject the null hypothesis that the means are the same, we followed this test with a post-hoc Tukey test. A Tukey test calculates pair-wise 95 percent confidence intervals of the difference between the means. If zero is inside this interval, we can conclude the pair does not show enough evidence to claim there is a difference in the means. The Tukey test confirmed that the difference in each pairwise comparison of our means were all statistically significant. Residual analysis was also performed on the data to verify that ANOVA was an appropriate test for our dataset.

The mean differences show in Figure 1 are, of course, not as dramatic as the worst possible differences in individual cases such as the differences reported in the Robinson case [1]. They indicate more wide-spread underlying differences. For example, for non-contributors, LRmix is more exclusionary than FST especially in its default settings while EuroForMix is more inclusionary. The mean differences are smaller for contributors, but the trends are the same. These differences suggest a natural question: why are we not more regularly requiring that the output of these different trusted black-box systems be compared. If the court system would trust the output of a number of different software packages, but they don't agree, that could be one way to provide examples to incentivize debugging and iterative improvement. Some difference may be due to differences in the underlying models, but some differences may be due to errors.

In Figure 2, we focus on one system, LRMix, and evaluate how changes in two key parameters, drop-out rate and theta, impact the results. As we have seen, drop out rate is related to the liklihood of drop out errors. The theta parameter is related to the amount of genetic diversity in a population. [1]

Once again, the Y-axis shows the mean likelihood ratio (log scale) produced for both true contributors (mean of approximately 1200 total LR results for true contributors) to test samples and for a set of known non-contributors (mean of approximately 336,000 total LR results for known non-contributors).

Figure 2 illustrates that varying the drop-out rate and theta both have substantial impact on the LR results. Higher drop-out rate yields more inclusionary results, while higher theta values yield more exclusionary results. Notice that the contributor line crosses the Inclusion-Exclusion boundary in both cases. Thus, changes in these parameters can mean the difference between inclusion and exclusion! Residual analysis and ANOVA were performed on this data to confirm that the differences are statistically significant.

In the case of the theta parameter, it is crucial to consider the potential for disparate impact. Theta is designed to account for differences in the genetic diversity among populations. Lower theta values would be appropriate for a population with high genetic diversity while higher theta values would be appropriate for a population with lower genetic diversity. For example, some indigenous

[1] The results of experiments varying additional parameters with FST, LRMix and EuroForMix, as well as additional materials related to this work, are available at https://afsweb.clarkson.edu/projects/cjsoftware/.

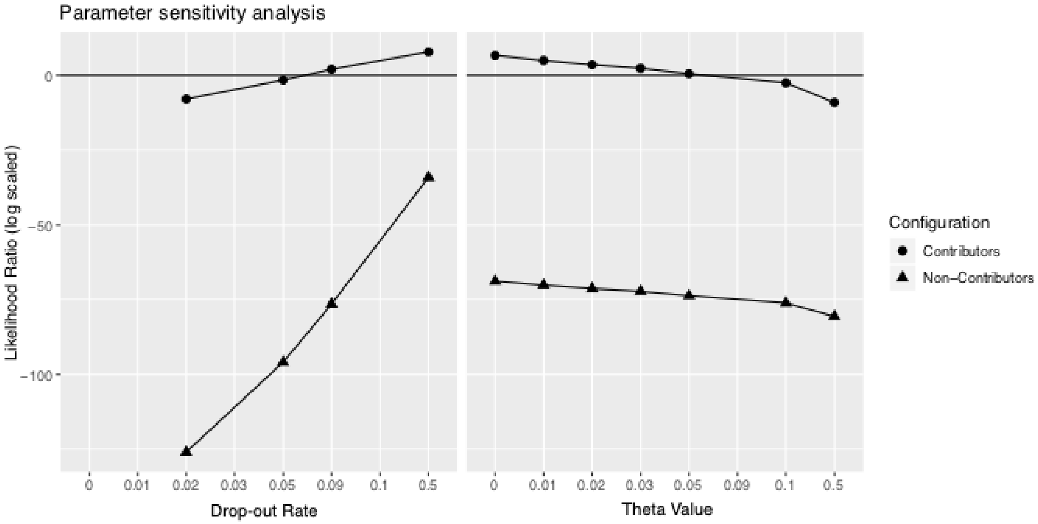

Figure 2: Y-axis shows the mean likelihood ratio (log scale) produced for both true contributors to test samples and for a set of known non-contributors. We test LRMix with four different settings for the drop-out rate parameter [LEFT] and seven different settings for the theta parameter [RIGHT]. Results above 0 (log 1) are inclusionary and results below 0 are exclusionary.

communities may have lower genetic diversity because they live in a smaller, more isolated community.

Allele frequencies used in PG software are often grouped into highly questionable categories such as Black, Caucasian, Hispanic, and Asian. As defined in this way, the Caucasian subpopulation has relatively high genetic diversity and thus for a constant theta value would have a lower false inclusion rate than some other populations and may therefore be less vulnerable to being falsely accused. Similarly, a theta value set based on testing for the Caucasian subpopulation could lead to false inclusions for other smaller, potentially vulnerable, minority populations. This is especially concerning given the tendency for critical software systems to be tested only on majority groups [2]. If these systems are not being actively tested against diverse sub-population benchmarks, there is a clear risk of disparate outcomes along the lines of protected categories such as race and ethnicity. In one notable case, PG systems were used as part of an investigation into an assault by a group of 8 Hasidic men. Concerns were raised about the ability of PG software to accurately distinguish among members of this more genetically insular population [6]. PG software testing with a sample containing DNA from 8 Hasidic men might incorrectly include another Hasidic man whose DNA was not actually part of the sample.

The presentation of LR results in court is also not as simple as just inclusion or exclusion. There are scales that convert LR values into a textual ratings such as "Exclusion," "Limited Support," "Moderate Support," "Strong Support," and "Very Strong Support." In Figure 3, we show the impact on all 3 systems of two different standards for mapping LR results onto verbal descriptions. On the x-axis, we show the OCME verbal standard and on the y-axis the SWGDAM verbal standard. These verbal standards dictate how a forensic analyst interprets the results and how they will represent the result when evidence is presented in court. We performed a Chi Squared test of independence. The null hypothesis, in this case, is that the proportion of data binned into each category, exclusion, weak inclusion, etc., will be the same for the OCME verbal standards and the SWGDAM verbal standards. A low p-value infers a statistically significant difference in how the results would be testified in court. This test was performed for each configuration, and for each test the p-value was less than 0.001, confirming that the same results would be categorized differently based on a lab's choice of standard, and therefore presented differently in court.

4 RECOMMENDATIONS

In total, we have documented substantial differences in the results generated by three PG systems as well as the large impact of varying parameters whose contributions are often not clearly appreciated. We have also seen the impact of different verbal standards on how results are presented. Despite the fact that these systems do the same job, there has been relatively little effort to standardize their behavior. We recommend that multiple independent PG systems be regularly compared for real casework. If we would trust the output of any of these systems, what does it say when they do not agree? By not regularly comparing these systems to each other on real case work, we are losing a key opportunity to regularly surface flaws and incentivize iterative improvement. We also recommend that agencies such as the National Institute of Standards and Technology (NIST) establish standards for key aspects of the development and use of PG systems including key parameters (both their meanings and their granularity of control), input file formats, output file formats, and population frequency definitions. We also recommend that more information be provided in court about the way in which the settings for key parameters are determined and whenever the setting is uncertain, how the software results would change if these parameters were set differently. An important question to ask is who in the larger socio-technical system is best able to set each parameters. Are some parameters best set by developers for all

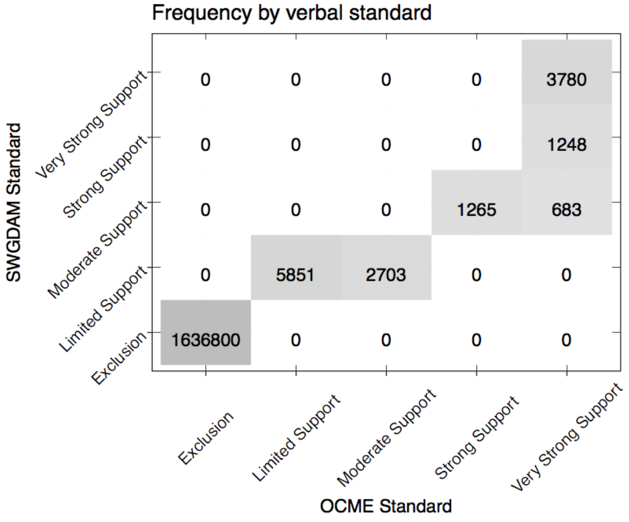

Figure 3: We bin the data from all three systems shown in Figure 1 by the OCME verbal standard and the SWGDAM verbal standard, two scales that convert LR values into a textual rating. The x-axis is the OCME verbal standard and the y-axis is the SWGDAM verbal standard. Deviations from the diagonal show differences in how the results will be presented/interpreted.

cases? Will the users (forensic analysts in the case of PG software) have the information they need to set key parameters on a case by case basis?

We have discussed the potential for disparate impact on along the lines of protected categories such as race and ethnicity. We highly recommend that developers of any critical software system actively test against diverse sub-population benchmarks. Legislation is likely to be required to achieve this. In the United States, the Algorithmic Accountability Act would, if passed, require companies to study and fix flawed computer algorithms that result in inaccurate, unfair, biased, or discriminatory decisions is one example of possible legislation in this space[14]. Because it is difficult to anticipate all possible sub-populations of interest, neither terms of service nor other legislation should be allowed to prevent third-party testing for disparate impact or the publication of the results of such testing.

We strongly recommend adversarial testing done by groups who will be rewarded when they discover flaws. When validation studies are only performed by the developers, they are likely to focus on demonstrating the effectiveness of the system rather than on aggressively identifying problems. Adversarial testing is important to counteract the tendency to sweep errors under the rug when they are found or to put in an inappropriate fix to make the problem go away as we saw in the FST case [8].

Further, we recommend targeting the procurement phase of software whenever public funds are used to purchase software. Procurement policies should require or at least give substantial credit for products that include pro-transparency factors. Such factors could include open-source software, access to software engineering artifacts including bug tracking/change log databases, internal testing plans and results, software requirements and specifications, hazard and risk assessments, design documents, etc. Ideally, developers or third parties would offer bug bounties or other funding streams to incentivize third party testing.

It is worth specifically considering whether public software should more generally be mandated to be open source. It is interesting that for PG software this is much more common in Europe. In the United States, it is not clear that there is the political will to require this. Even defense experts in criminal cases are regularly denied access to source code and system details under protective order when software vendors claim trade secret protection [13]. There are tools and strategies for auditing software systems beyond source code access. It is worth mentioning that for software systems with substantial AI/machine learning components, it may be even more important to have information about the training data used that to have access to the source code. Alternate tools and strategies for algorithmic accountability and explanation are even more essential in these cases [5, 10]

Most PG systems are designed with casework in mind. Many systems have no native ability to batch-process multiple evidence samples or compare a single evidence item to multiple reference profiles (e.g. a set of non-contributors or an offender DNA database). While it is possible to modify source-available systems to enable batch-processing, any third-party modification risk introducing defects and require further software validation. APIs, or at least command line interfaces, could allow for easier batch-processing tasks in both casework and research settings. Requiring these during the procurement phase would be an important advance.

Many important decisions about the lives of individuals are made with software chosen outside the context of a public procurement process. In these environments, there are even more hurdles to the type of adversarial third party testing necessary to reveal problems. In an environment where the terms of service of many key platforms seek to prevent such testing. We should recommend lobbying for provisions in legislation such as the Computer Fraud and Abuse Act and the Digital Millennium Copyright Act to specifically permit testing aimed at identifying disparate impact and other possible violations of laws regarding fair practices in areas such as hiring, housing and credit.

5 CONCLUSION

Software is increasingly used to direct and manage critical aspects of all of our lives from how we get our news, to how we find a spouse, to how we navigate the streets of our cities. Beyond personal decisions, software plays a key role in regulated areas like housing, hiring and credit, as well as major public functions such as criminal justice and elections. It is easy for there to be unintended defects with a large impact on the lives of individuals and society as a whole. Bugs enter the systems at design time, during implementation, and during deployment. Additional problems occur in the way software systems are used in individual cases. Testing against diverse sub-population benchmarks is especially critical. Generally, we must ask what in the larger socio-technical decision-making system will incentivize debugging, iterative improvements and adherence to the laws that protect the rights of individuals in these black-box systems. Without a credible plan to incentivize debugging and iterative improvement, it is irresponsible to deploy software in critical application areas.

In this paper, we have used criminal justice software, specifically probabilistic genotyping (PG) software, as a concrete example of black-box systems that can be unresponsive to the interests of individuals about whom big decisions are being made. We describe how PG software systems, designed to do the same job, produce different results and can have a disparate impact on different racial/ethnic groups. We also discuss the impact of these differences on how the results are presented in court. We proposed concrete changes to the socio-technical decision-making processes surrounding the use of PG software that could be used to incentivize debugging and improvements in the accuracy, fairness, and reliability of these systems and how these lessons could be applied beyond PG software to the multitude of critical software systems impacting the lives of individuals today.

ACKNOWLEDGMENTS

The Brown Institute at Columbia University and Stanford University generously supported this work through a 2018-2019 Magic Grant. Thank you to the other members of our Magic Grant Team: Nathan Adams, Dan Krane, Surya Mattu and David Madigan. Thank you also to the large team surrounding the project including Clinton Hughes, Gabriel Ferrante and many others. The McNair Scholars program at Clarkson University supported Mariama Njie's work in Summer 2018 and 2019. We also thank the members of the Clarkson Open Source Institute for their support.

REFERENCES

[1] 2017. Post-conviction hearing in the matter of People v. Robinson, (ordering DNA testing on certain samples). 147 A.D.3d 784 (2d Dept. 2017).

[2] Joy Buolamwini and Timnit Gebru. 2018. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*. 77–91.

[3] John M Butler, Richard Schoske, Peter M Vallone, Janette W Redman, Margaret C Kline, et al. 2003. Allele frequencies for 15 autosomal STR loci on US Caucasian, African American, and Hispanic populations. *Journal of forensic sciences* 48, 4 (2003), 908–911.

[4] P. Garofano, D. Caneparo, G. D'Amico, M. Vincenti, and E. Alladioa. 2015. An alternative application of the consensus method to DNA typing interpretation for Low Template-DNA mixtures. *Genetics Supplement Series* 5 (2015). https://doi.org/10.1016/j.fsigss.2015.09.168

[5] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna M. Wallach, Hal Daumé III, and Kate Crawford. 2018. Datasheets for Datasets. *CoRR* abs/1803.09010 (2018). arXiv:1803.09010 http://arxiv.org/abs/1803.09010

[6] Lauren Kirchner. 2017. Thousands of Criminal Cases in New York Relied on Disputed DNA Testing Techniques. https://www.propublica.org/article/thousands-of-criminal-cases-in-new-york-relied-on-disputed-dna-testing-techniques. (September 2017).

[7] Steven P Lund and Hari Iyer. 2017. Likelihood Ratio as Weight of Forensic Evidence: A Closer Look. *Journal of Research of National Institute of Standards and Technology* 122, 27 (2017). https://doi.org/10.6028/jres.122.027

[8] Jeanna Matthews, Marzieh Babaeianjelodar, Stephen Lorenz, Abigail Matthews, Mariama Njie, Nathaniel Adams, Dan Krane, Jessica Goldthwaite, and Clinton Hughes. 2019. The Right To Confront Your Accusers: Opening the Black Box of Forensic DNA Software. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society (AIES '19)*. ACM, New York, NY, USA, 321–327. https://doi.org/10.1145/3306618.3314279

[9] Adele A Mitchell, Jeannie Tamariz, Kathleen O'Connell, Nubia Ducasse, Zoran Budimlija, Mechthild Prinz, and Theresa Caragine. 2012. Validation of a DNA mixture statistics tool incorporating allelic drop-out and drop-in. *Forensic Science International: Genetics* 6, 6 (2012), 749–761.

[10] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *CoRR* abs/1602.04938 (2016). arXiv:1602.04938 http://arxiv.org/abs/1602.04938

[11] Supreme Court Kings County New York 2015. People v. Collins. 15 N.Y.S.3d 564 (Sup.Ct. Kings Co. 2015).

[12] H. Swaminathan, M. Qureshi, C. Grgicak, K. Duffy, and D. Lun. 2018. Four model variants within a continuous forensic DNA mixture interpretation framework: Effects on evidential inference and reporting. *PLOS One* 13, 11 (2018). https://doi.org/10.1371/journal.pone.0207599

[13] Rebecca Wexler. 2018. Life, liberty, and trade secrets: Intellectual property in the criminal justice system. *Stan. L. Rev.* 70 (2018), 1343.

[14] R. Wyden, C. Booker, and Y. Clarke. 2019. The Algorithmic Accountability Act.

A APPENDIX: FST AND THE OCME VALIDATION STUDY

The NY State Commission on Forensic Science approved FST for use in casework based on a validation study designed and conducted by Office of the Chief Medical Examiner (OCME). The validation study underlying FST consisted of 439 two- and three-person mixtures of varying quantities of DNA and contributor proportions, genotyped using both High Copy Number (HCN) and Low Copy Number (LCN) protocols. Since these mixtures were created in a controlled laboratory setting, their true contributors and known.

The samples were constructed based on single-source blood and cheek swab samples of known origin as well as from items handled by multiple individuals, such as a computer mouse or a pen. Some, but not all, of the touched items were cleaned with bleach and ethanol prior to handling. Despite this pre-cleaning step, it is interesting to note that some samples still contained DNA that did not belong to any of the deliberate contributors.

OCME evaluated all 439 mixtures in comparison to their known contributors and a set of 1,246 non-contributors. The non-contributor set consists of genotypes developed from OCME morgue bodies and a national data set [3]. Allele frequency rates were established for NYC by OCME through genotyping morgue bodies. OCME developed a subset of these genotypes at only thirteen of the fifteen loci used by FST, simulating genotypes for the remaining two loci. Sub-populations were grouped by self- or OCME-reporting into African-American, Asian, Caucasian, and Hispanic categories. The lab removed information on the races of the donors to the mixtures, though in publications they do claim that the mixtures represent the diversity of New York City [9, 11].

OCME originally wanted to validate FST for four-person mixtures and additional four-person mixtures were generated during the study, but ultimately FST was not validated for the evaluation of four-person mixtures [9]. OCME never published the validation data set but did produce it in 2012 pursuant to an agreement reached after litigation in the case People v. Collins. It was produced in printed form, then scanned and partly transcribed by the defense team.