



## Can Open-Source Fix Predictive Policing? Anti-Racist Critical Code Studies Approach to Contemporary AI Policing Software

Sarah Ciston <sarahciston\_at\_gmail\_dot\_com>, University of Southern California  <https://orcid.org/0000-0001-9456-9537>

Zach Mann <zmann\_at\_usc\_dot\_edu>, University of Southern California  <https://orcid.org/0009-0003-8576-1908>

Mark C. Marino <markcmarino\_at\_gmail\_dot\_com>, University of Southern California  <https://orcid.org/0000-0003-2034-3433>

Jeremy Douglass <jeremydouglass\_at\_gmail\_dot\_com>, University of California Santa Barbara  <https://orcid.org/0000-0001-7798-8801>

### Abstract

Technology watchdogs and technoculture critics have discussed predictive policing software at an abstract level or have tried to reverse engineer its blackboxed code. In this paper, we use the methods of Critical Code Studies, media archaeology, and software studies more broadly to analyze CivicScape predictive policing software, published online, albeit partially. Working from an anti-racist approach, we examine how the CivicScape code calculates which neighborhoods to recommend for heavy policing. Our reading demonstrates what code analysis can add to the analysis of such software and makes a case for the public release of all legislative operational source code for scrutiny under the principles of the Freedom of Information Act.

Over the past decade, governments have begun using machine learning software to guide decisions, including how to allocate police resources. In response, scholars and advocates of anti-racism have called for a critical examination of the algorithms used to make these decisions. In theory, the use of software in public policy seems to facilitate resource allocation decisions by offering computational efficacy and precision with relatively low-cost processes [Yuill 2019]. However, in practice, the application of these software tends to circumvent public scrutiny. The code for many tools of predictive policing are in “black boxes”, or hidden from public scrutiny, something Sun-Ha Hong describes as “prediction as a source of opacity” [Hong 2015]. Since algorithms like those used in predictive policing are already being used to govern our lives, we who are policed by them — or for whom policing decisions are made — should have access to their code and its workings. Previous scholars have approached predictive policing software by discussing its related data, both inputs and outputs (e.g., [Richardson, Schultz, and Crawford 2019] and [Sankin et al. 2021]). In contrast, we selected one of the only open-source examples and used the methods of Critical Code Studies (CCS) to close-read the source code itself. What follows is an analysis based on some preliminary discoveries.

1

By definition all machine learning tools involve some form of discrimination (judgments based on weighted values) and even, arguably, prejudice. However, by examining one example closely, we hope to better understand and perhaps challenge the decisions made in the creation of this discriminatory tool. We also hope to press this examination beyond the narrower critique of ethics, in alignment with calls from scholars such as Ruha Benjamin and Matthew Fuller, into the realm of politics and State control. Such an expansion, Fuller argues, would allow for a broader critique of power in code, whether it be governmental or corporate power, as it processes subjects [CodeFest 2021].

2

Our method, Critical Code Studies, names the interpretation of the extra-functional significance of computer source code. Rather than serving as an end to an investigation, code is the starting point for a larger discussion of technoculture [Marino 2006] [Marino 2020]. The goal is not to find a secret key to the software in the code but instead to explore the signs of code for evidence and insights into how the code works, how it was made, and, more importantly,

3

how it communicates and, in this case, polices civic spaces. In this paper, we attempt to demonstrate the kinds of decision-making processes, or computational policies, that can be brought to light by reading the code. Furthermore, we argue that the source code for these systems should be subject to existing freedom of information rights that apply to other tools of governance.

## Anti-Racist Critical Code Studies<sup>[1]</sup>

The term *anti-racism*, as framed by Ibram X. Kendi, goes beyond the concept of *non-racist* to call for participatory action toward systemic change beyond anti-discrimination. Kendi argues, “We’re either supporting policies that are leading to racial inequities and injustice...or we’re supporting policies and pushing policies that are leading to justice and equity for all” [Kendi 2019]. Our investigation builds on work begun during the 2021 Anti-Racist Critical Code Studies Reading Group, which was inspired by Kendi and scholars working on algorithmic injustice at the intersections of critical race theory and algorithm studies. Algorithms are used in decisions to terminate individuals’ Medicaid and food stamps, to determine who is put on the No Fly List, and who is hit in drone strikes. These tools operate within a prison-industrial complex that disproportionately imprisons black and brown bodies,<sup>[2]</sup> leaning heavily on biased input data and faulty decision-making systems which leave marginalized communities further at risk, all while offering an air of detached rationality. Since policing shifted in the late 20th century to focus on crime *prevention* — through stop and frisks, neighborhood watches, early hotspot mapping, and so forth — such practices have amplified the pre-existing tendency for the government to racially profile urban spaces. When these practices get outsourced to software, the biases do not stop; they just become technologized, automated, and further hidden from accountability. Many cities have started banning the municipal use of predictive policing entirely, and yet, these algorithms will likely only become more ubiquitous.

4

## Previous Scholarship on Predictive Policing

Predictive policing software rely on a few common assumptions: the near repeat hypothesis, which presupposes that an event occurring in one location is evidence that the same event is more likely to reoccur in the same location than elsewhere; broken-window policing, a strategy which relies on correlations between major crimes and minor factors such as jaywalking, vandalism, and building neglect; and the notion that police presence necessarily prevents crime. Peter Polack challenges all of these premises, claiming that identifying “hot spots” from crime history is a self-fulfilling prophecy because increasing police presence in an area also increases the reporting of minor offenses for those areas (and minor offenses already make up 80 percent of the data these software rely on) [Polack 2020]. Instead of fixing windows or increasing community services, predictive policing algorithms process causal data and turn it into symptoms. Police might even be encouraged to look for suspicious behavior where there is none. These self-fulfilling loops, the epistemological core of predictive policing, are then formalized in software and rendered opaque [Kaufmann, Egbert, and Leese 2019].

5

Meanwhile, broken-window policing has long been proven to be compromised by the biases (which may be related to the intersections of race, gender, ability, class, and other differences) of those doing the reporting, including the police, individual citizens, and neighborhood groups via apps such as NextDoor. This is not, as many narratives suggest, “big data” in action — there is little novelty in what predictive policing software provides police departments [Sandhu and Fussey 2021]. Algorithms are only as good as their designers, their data, and the police who use them. Data, when selected for a use-case already in mind, is always constructed, never raw or objective, and the relationship between police and data is always a complicated one. It begins as subjective, but when fed into algorithms it is delivered back to us as a distant objectivity — or, as Sun-ha Hong terms it, “technological rationality” [Hong 2015].

6

Nick Lally points out, additionally, that these predictive policing algorithms might be trained on location-specific data once installed in a police department, but the base theories they rely on — and the shapes of the crime patterns they assume — were developed elsewhere, often in the United Kingdom [Lally 2021]. Some of the details of the software are hidden from the police to protect the rights of the citizens, and police themselves reportedly tend to favor their own “instincts” over the analysts’ recommendations, meaning there is a double-blind exchange occurring between the “blue

7

wall” of the police department and the “black box” of the software that leaves no one accountable. This functional modularity compounds with the tendency of software, as Tara McPherson has pointed out, to operate by logics which resist examination across more than one set of social contexts [McPherson 2013]. One kind of software might be used by a police department as one of many tools, collected into a single dashboard with similar services like ShotSpotter Missions (for gunshot data recording). Likewise, these software might be used to influence other municipal decisions (such as parade routes and construction zoning). Because software behave modularly, it is often difficult to trace their influence.

Some researchers have attempted to quantify the success or failure of predictive policing software, and while there is some suggestion that residential burglaries can be reduced through the use of such software, there is little to no empirical evidence that predictive policing has improved policing or civic life [Meijer and Wessels 2019]. Systems which rely on patterns only capture offenses that follow the rules upon which that algorithm relies (and arguably maintains as the norm) [Kaufmann, Egbert, and Leese 2019]. One additional challenge is the fact that the developers and intended users of these software pay more attention to correlation than causality; for this reason and the other complications above, when we consider whether such predictions might be “bad”, we are less interested in its accuracy after the fact than the ways in which the code formalizes certain conservative or aggressive policing strategies and behaviors, the assumptions made behind the decision-making which then feed back into the culture of policing, and the normality of crime.

Other scholars challenge the preemptive logic behind the implementation of predictive policing in the first place. While there may be value in determining the relationship between historical and environmental factors and the probability of crime, such as incidents of dog bites that occur in a playground next to a dog park, when these calculations are removed from their contexts, as they are in machine learning, and are used to determine levels or quantities of policing, they fuel and seem to justify “possibilistic” or “paranoiac” thinking. Thus, they transform from analysis of the past to active conjecture [Egbert and Krasmannb 2020]. Predictive policing algorithms produce a desired future: that is, a future desired by the police, or what Bonnie Sheehey calls “temporal governmentality” [Sheehey 2019]. Sun-ha Hong and Piotr M. Szpunar argue that such software exploit the uncertainty of the future as a Trojan horse for inventing a future based on preset ideologies, a future which becomes retroactively legitimized by these so-called “predictions”. Selective future-oriented truthmaking becomes “laundered” when concrete actions (like police deployment) are taken on behalf of algorithms too complex to be understood, with margins of error too messy to be measured. Unprovable judgments are thus transformed by these loose thresholds of proof into “justificatory cover for state power” [Hong and Szpunar 2019, 315]. They shape the way we think about our cities often without us knowing it.

## Choosing an Object: Predpol versus CivicScope

For our purposes, the search for an object of study for predictive policing code presented a problem. It is hard to understand why more of this type of software is not available for public exploration. On the one hand, the software was developed by for-profit corporations. On the other hand, the effects of this code are government regulatory mechanisms, no less than law. We would advocate that all such code be made public following the model of the Freedom of Information Act as well as the larger push for government transparency. As it stands, even though such software is used by publicly funded governmental bodies, the code for this software is often black-boxed or kept from view, in part to protect the proprietary code for the profit of the companies that build them, and perhaps also to protect the processes from scrutiny. Consequently, those who wish to study the algorithms have had to rely on studies of the inputs and outputs of the software, exemplified by the ProPublica exploration of Northpointe's sentencing software [Angwin, Kirchner, and Surya 2016]. In that case, Northpointe gave ProPublica “the basics of its future crime formula”. While approximations may be necessary for the study of code that is inaccessible, we also acknowledge the limitations of such analyses.

Though predictive policing existed in rudimentary forms as early as the implementation of the Compstat system in New York in the 1990s [Benbouzid 2019], emerging simultaneously with so-called “evidence-based policing”, innovations in predictive policing spiked after 9/11, with the terminology “predictive policing” appearing as early as 1997 [Schellenberg 1997]. Today, there are a number of different pieces of proprietary software that have contracts with police departments:

PredPol, CivicScope, Shotspotter Missions (formerly HunchLab), Palantir, Rutgers University's RTM Diagnostics, Carnegie Mellon University's CrimeScan, and counting [Human Rights Watch 2018].<sup>[3]</sup> For our initial exploration, we considered the most (in)famous of the bunch. PredPol, short for Predictive Policing (and since renamed Geolitica), essentially marked the beginning of active predictive policing when it was adopted by the police departments of Santa Cruz and Los Angeles in 2011, in partnership with UCLA. It also remains the leading such software in the U.S. However, Predpol has, with legal protection, refused to release the details of its software. In 2016, public pressure convinced them to publish a description of how their algorithm worked. Kristian Lum and William Isaac then reverse-engineered the software, using synthetic data from Oakland to project how it *might* look in action. But this model was only ever a simulation.

In 2017, CivicScope became the first predictive policing company to make its algorithm public [dkg 2017]. They released their source code and input variables onto Github, based on the premise that transparency would lead to a more accurate and less biased system. Although the simulated PredPol software was well-documented and seemed to operate in the same way, we ultimately decided that CivicScope was the better choice for our analysis. While a study of the PredPol replica may yield many of the same insights, an approximation can introduce differences in the software that, while perhaps minor, may have larger impacts on its significance. An exploration of CivicScope, by contrast, would allow us the opportunity to explore the actual tokens, the literal signs used by the programmers and circulated within the program.

12

## CivicScope: A Case Study

CivicScope is born out of the fantasy of a police “moneyball”,<sup>[4]</sup> drawing upon the approaches popularized by the film of that title about the use of data in baseball team management. The appeal of the *Moneyball* story is the seemingly magical way in which an analyst can identify unexpected relationships between data and then capitalize on them. Predictive policing software trades on similar promises. The underlying presumption of the software is that correlated data can be used to identify neighborhoods in need of increased police presence, which would then lead to a decrease in crime. The fantasy of machine learning and artificial intelligence is the notion that software can find these correlations without the intervention of humans.

13

One of the founders of CivicScope, Brett Goldstein, is an ex-police officer for the Chicago Police Department. Goldstein became a beat cop in 2006, but he had previously been the director of information technology at OpenTable [Brustein 2017]. After only a year working a beat, he started working on predictive policing models.<sup>[5]</sup> Goldstein later brought his work to the private company Ekistic Ventures, and, by 2017, CivicScope was used in nine cities, including Chicago and Philadelphia. For about a year, it received positive media coverage for its commitment to transparency and the company's self-reported success with predicting opioid overdoses using 911-call data [Brustein 2017]. Since then, CivicScope has largely disappeared from the public view, but it continues to be used in policing, for example, in Camden, New Jersey [Noone 2021].

14

There are two main approaches to predictive policing: deciding *whom to police*, through software that tracks probabilities that individuals will be perpetrators or victims of a crime, or *where to police*, through software which targets locations on a map. The former type, like the Strategic Subjects List (SSL) used by the Chicago Police Department, essentially functions as automated “profiling”, placing certain people on “heat lists” in order to be surveilled by the police. However, like PredPol and most new predictive policing services, CivicScope is part of the second class, targeting places, not people. It uses an algorithmic model called ETAS (Epidemic Type Aftershock Sequence), based on previous tools used to predict earthquakes, which converts historical data into geographic probability distribution. The ETAS algorithm calculates the probability that criminal behavior might occur in certain areas of a map due to (1) that area's characteristics, and (2) whether or not certain types of crime recently occurred there (similar to “aftershocks”). This is done with machine learning: a series of neural networks are trained on past crime data in order to forecast which areas are more likely to see future crime, which is distilled to a corresponding “risk score” for each location. These risk scores are delivered to analysts in the police department who then prioritize police presence in locations identified as “hot spots”. Renata M. O'Donnell critiques predictive policing software, writing, “Unlike a free-thinking racist neighbor,

15

programmers create predictive policing algorithms specifically for a state actor and feed those algorithms data that is generated by that same state actor” [O’Donnell 2019, 578]. Our investigation into the code of CivicScape’s public release suggests that characterization is not accurate.

CivicScape uses crime data (mostly of violent and property crime), community input (namely 311 reports), census tracts (for mapping), and weather forecasts. Because crime is statistically rare, CivicScape uses “downsampling” to train its models; like many machine learning approaches to policing, they artificially increase crime frequency (so it makes up 50 percent of the training data). CivicScape also claims to use machine learning to test for its own biases (filtering out data with missing information and training on random subsets of data to identify outliers), though the extent to which this testing exceeds the usual best practices for machine learning is unclear.

16

## CivicScape: Reading the Code

The question remains: What can a close reading of the code of CivicScape reveal about the software, or, more importantly, to CCS and in this particular moment in history, what can an examination of this example of predictive policing software reveal about the cultural logic of artificial intelligence used in law enforcement.

17

Reading the CivicScape code, we argue that this machine learning software is a mechanism of misdirection that reenacts many of the issues of predictive policing software raised above. First, the opacity of the process seems to render the decision-making inaccessible. The process is trained on existing data before weighing new data in ways that cannot be easily identified, leaving reviewers to speculate about the inner workings of the software based on inputs and outputs, which is confounded by the unseen associations generated in machine learning, as seen in the problem commonly known as “explainability”.<sup>[6]</sup> Second, the process involves a self-confirming hypothesis that certain factors can be used to identify “bad neighborhoods”, or civic spaces which require increased policing, which in turn leads to the observation of more crimes. Third, the version of the software made public for review does not include key data such as records of crimes and data on known criminals. Instead, we have public records without the crucial crime data, so we cannot know the actual process. Fourth, the measure of the software’s validity could only be in its confirmation of pre-existing judgments of the neighborhoods. Lastly, and perhaps most importantly, given the long shadow of institutional racism, the disproportionate judgment of non-white, BIPOC (black, indigenous, and people of color) offenders make the inclusion of such data merely another mechanism for the State policing of black and brown persons. Notably, CivicScape excludes low-level drug crimes from its data due to reported bias, but it still relies on community and police reporting to gather its data. As Lum and Isaac concluded in their study of PredPol, such software are susceptible to the notion of “garbage in, garbage out”. CivicScape’s own documentation even repeats this mantra.<sup>[7]</sup>

18

Although we cannot include a full review of the code in the small span of this paper, we take one portion of it as a demonstration of a CCS reading. For our reading, we are focusing on a sample of the code from the “Training\_and\_Testing” folder of CivicScape’s GitHub, specifically the code that processes data from a 311 telephone line in Philadelphia. Philadelphia’s 311 number, or Philly311, is a line “for non-emergency inquiries. Requests for service”.<sup>[8]</sup> In addition to calling the phone number, citizens can make reports through mobile and web apps. The Philly311 phone app allows anyone to form neighborhood watch groups, or report broken street lamps or abandoned cars in the moment by taking a photo and uploading it to the app with their location services on. Dozens of 311 reports are added per hour, and you can view a real-time map of the reports. CivicScape uses this data in order to create models of neighborhoods to determine the relative need for police presence.

19

By drawing upon these 311 calls, CivicScape follows the logic of city governments working to repair windows in order to stop crime, except, in this case, rather than proactively looking for fractured windows, the system processes received complaints to determine where more policing is needed. According to the broken windows theory, first introduced by Kelling and Wilson, “One unrepaired broken window is a signal that no one cares, and so breaking more windows costs nothing” [Kelling and Wilson 1982]. However, the authors of this influential theory claim that the panhandler is “the first broken window”. They write, “Muggers and robbers, whether opportunistic or professional, believe they reduce their chances of being caught or even identified if they operate on streets where potential victims are already intimidated by prevailing conditions”. Despite many challenges to the merits of these claims, this model of CivicScape uses such data

20

to profile neighborhoods.<sup>[9]</sup> When CivicScape trains its algorithm on 311 complaints, it is compiling a whole litany of broken windows, indirect indicators of a neglected neighborhood or, to put it another way, signs that a neighborhood is ripe for crime.

Consider the following passage of SQL code:

21

```
create view t311
FROM
(
select
case when cell_id is not null then '4' else '4' end as city
, cell_id
, date_trunc('hour', "Requested Date/Time") as hr
, case when t311_all."Service Name" = 'Graffiti Removal' then 1 else
0 end as graf
, case when t311_all."Service Name" = 'Illegal Dumping' then 1 else
0 end as illdumping
, case when t311_all."Service Name" = 'Maintenance Residential or
Commercial' then 1 else 0 end as bldgmaint
--, null as streetsw
--, null as electrical
, case when t311_all."Service Name" = 'Sanitation / Dumpster
Violation' then 1 else 0 end as sanitation
--, null as recycling
, case when t311_all."Service Name" = 'Street Trees' then 1 else 0
end as tree
, case when t311_all."Service Name" = 'Traffic (Other)' then 1 else
0 end as traffic
, case when t311_all."Service Name" = ' Vacant House or Commercial'
then 1 else 0 end as vac_bldg
, case when t311_all."Service Name" = 'Abandoned Vehicle' then 1
else 0 end as vac_vehicle
, case when t311_all."Service Name" = 'Street Light Outage' then 1
else 0 end as stlghts
, case when t311_all."Service Name" = 'Alley Light Outage ' then 1
else 0 end as alleylghts
--, null as potholes
, case when t311_all."Service Name" = 'Rubbish/Recyclable Material
Collection' then 1 else 0 end as garbage_pickup
--, null as rodent
--, null as sidewalk
```

CivicsScape first uses create view t311 to create a virtual table from the 311 call data.

22

```
FROM
(
select
case when cell_id is not null then '4' else '4' end as city
, cell_id
, date_trunc('hour', "Requested Date/Time") as hr
, case when t311_all."Service Name" = 'Graffiti Removal' then 1 else
0 end as graf
, case when t311_all."Service Name" = 'Illegal Dumping' then 1 else
0 end as illdumping
, case when t311_all."Service Name" = 'Maintenance Residential or
Commercial' then 1 else 0
end as bldgmaint
```

Here is the first set of notifications CivicScope tabulates: graffiti removal, illegal dumping, and maintenance. If there was a record of a call on any of these items, the program will register it with a value of one. While the first two of these notifications may relate to crimes, the third one is ambiguous and seems more closely related to “broken windows”, with only an implied correlation to criminal activity, presumably as a setting or signal of low oversight as we discussed earlier.

```
--, null as streetsw
--, null as electrical
```

Notice that neither street sweeping calls nor electrical calls are listed, so there is some exclusion happening here. In other words, not every type of call to 311 is included in this training data. Apparently, there are limits to what counts as a broken window. The next set offers the reset of the data being included or excluded.

```

, case when t311_all."Service Name" = 'Sanitation / Dumpster
Violation' then 1 else 0 end as sanitation
--, null as recycling
, case when t311_all."Service Name" = 'Street Trees' then 1 else 0
end as tree
, case when t311_all."Service Name" = 'Traffic (Other)' then 1 else
0 end as traffic
, case when t311_all."Service Name" = ' Vacant House or Commercial'
then 1 else 0 end as vac_bldg
, case when t311_all."Service Name" = 'Abandoned Vehicle' then 1
else 0 end as vac_vehicle
, case when t311_all."Service Name" = 'Street Light Outage' then 1
else 0 end as stlghts
, case when t311_all."Service Name" = 'Alley Light Outage ' then 1
else 0 end as alleylghts
--, null as potholes
, case when t311_all."Service Name" = 'Rubbish/Recyclable Material
Collection' then 1 else 0 end as garbage_pickup
--, null as rodent
--, null as sidewalk
```

CivicScope will judge a neighborhood's policing needs based on sanitation or dumpster violations, street trees, traffic, vacant buildings, abandoned vehicles, and street or alley light outages, but not on potholes, rodents, or sidewalk complaints, presumably unless they involve something else from the former list. CivicScope then adds up all of these complaints as a representation of the area.

```

AS
select city, cell_id, hr, sum(graf) as graf, sum(illdumping) as
illdumping, sum(bldgmaint) as bldgmaint
, sum(sanitation) as sanitation, sum(traffic) as traffic,
sum(vac_bldg) as vac_bldg
, sum(vac_vehicle) as vac_vehicle, sum(stlghts) as stlghts,
sum(alleylghts) as alleylghts
, sum(garbage_pickup) as garbage_pickup
, sum(graf) + sum(illdumping) + sum(bldgmaint) + sum(sanitation) +
sum(traffic) + sum(vac_bldg) + sum(vac_vehicle)
+ sum(stlghts) + sum(alleylghts) + sum(garbage_pickup)
as t311Events
```

Note how each complaint is tallied in this equation and in particular the way each complaint, from illegal dumping to a complaint simply called “traffic”, are given equal weight (a value of 1) in this equation. Calls for vacated vehicles,

`sum(vac_vehicle)`, and street lights, `sum(stlights)`, are merely summed and added together. There are no multipliers to affect the way each complaint is weighted. At this point, we might begin to ask why some of these complaints are even being considered as part of the training data for identifying problem neighborhoods. Perhaps abandoned cars are a problem correlated with other factors related to crime or at least poverty or lax parking enforcement, but is traffic a sign of anything in particular other than perhaps population density or proximity to an urban center?

Stepping back, we can ask questions about the sources of the data. Who is making these calls? Are they made by entitled citizens who are used to having their complaints answered? Or are they made by exasperated citizens who are merely frustrated with the lack of response of civic leaders? Or are they made by citizens who have time to walk around and notice their neighborhood, which could be seen as a positive? And, most importantly, are any of these factors representative of the crimes that increased policing would counteract? Perhaps so in the case of complaints about graffiti, but what will an increased police presence do to alleviate traffic? In fact, it could have just the opposite effect, since increased police presence might further arrest the flow of traffic. Such a line of inquiry reveals that this training data does not directly correlate to the policing needs of the community but instead merely to an area where many complaints are made.

27

In our review of this code, one particular piece of data has caught our attention: complaints about street trees. In urban neighborhoods, communities have historically planted trees to provide shade and to beautify. These trees become objects of complaint when their roots break up the pavement or when their branches threaten power lines. However, what makes a troublesome street tree a necessary indicator of a neighborhood in need of increased police presence?

28

[10] Considering that question led us to the roots of the problem plaguing the implementation of this software.

## On Street Trees

How are street trees harbingers of bad neighborhoods? When examining the correlation between sidewalk or parkway trees and crime in Portland, one study was not conclusive [Donovan and Prestemon 2010]. Instead, their findings were

29

consistent with the principles underlying the broken windows theory: attributes of a neighborhood may provide information to criminals about the effectiveness of authority. Specifically, the presence of street trees may indicate that a neighborhood is more cared for and, therefore, a potential criminal is more likely to be observed by an authority [Donovan and Prestemon 2010].

Thus, the mere presence of street trees may suggest police presence is already high, though perhaps complaints about street trees offset this connection.

```
                , case when t311_all."Service Name" = 'Street Trees' then 1 else 0
end as tree
```

In this line, when the program encounters a notification for “Street Trees”, it logs the value 1.

30

The presence of the street trees in the training data raises the question: Who logs complaints against street trees? Who has the time not only to notice their need for tending but also to call in a complaint? What demographic feels confident such a complaint will be heeded? Does a 311 call even necessarily indicate a specific complaint, given that a call about a fallen tree branch and a call about a tree in need of tending would receive the same representation in this system? Moreover, at least one review of the Portland tree initiative suggests that the increased planting of street trees, all of which might yield calls for tending and care, correlates with a reduction in violent crime [Burley 2018]. It may very well be that these problematic trees reduce crime more than increased police presence. Perhaps CivicScape could be used to determine which neighborhoods require better arboriculture, rather than more surveillance.

31

The inclusion of calls about street trees in the training code for CivicScape reveals four important aspects of the software. First, a street tree is not a sign of criminal activity along the lines of vandalism. Second, the data does not

32



include any indication of the nature of the complaint about these trees. Third, it is unlikely that the people who call in the complaint are aware that they are identifying their neighborhood as a “bad neighborhood” or one in need of more police presence; those users in Philadelphia who are reporting 311 are not, when they are submitting their picture of an abandoned car, actively thinking about how they are constructing a dataset for a future algorithm, let alone a policing algorithm. This leads us to our fourth point, that those concerned about the status of their street trees are, by contrast, more likely to be interested in and part of a community that is concerned about its holistic wellbeing, from environmental health to safety to aesthetics, suggesting the relative health of that neighborhood overall. In many ways, an expression of concern about a street tree is the opposite of a complaint about vandalism. It reflects concern *for* as opposed to revulsion *at*. Such a distinction shows the way the training schema of CivicScope takes an uncritical approach to correlation that compounds our misunderstanding of the nature of our cities, the nature in our cities, and the nature of our citizens.

Street trees in our example act as a proxy for other data that CivicScope and other predictive policing may include. We already know that the decisions are being made. Take, for example, the very public pronouncement in the released files about minor drug offenses. In the notebooks on preventing bias, they write:

33

Minor marijuana possession cases are one of the most biased in terms of the discrepancy between the population who uses and is arrested for using drugs. The American Civil Liberties Union (ACLU) finds that marijuana use is roughly equal among African Americans and whites, yet African Americans are 3.73 times as likely to be arrested for marijuana possession. Overall data on drug use has shown that it is relatively representative of the general population, but it is more likely that drug sellers will face arrest and prison. There are no reliable surveys of drug selling, but given that people are most likely to buy drugs from someone of their same race, most researchers think that selling should be proportionate as well (Sentencing Project; Tonry et al.).

\n, [dkg 2017, line 760], evaluation\_notebooks/notebooks/PreventingBias.ipynb

This comment about minor marijuana offenses is designed to perform a sense of careful reflection on the kinds of data used in CivicScope. However, the 311 training call data shows more of a moneyball, “kitchen sink”-style approach, in which the system draws in any data it can access, such as calls about street trees, and then uses that to establish unexpected patterns of correlation.

34

Clearly, the creators of CivicScope are well aware of how their publicly released code will be analyzed, critiqued, and possibly even forked. They are also aware of the growing national support for the decriminalization of minor drug offenses as well as the correlations between enforcement of anti-drug laws and the mass incarceration of black and brown bodies. They explicitly address race later in that same notebook:

35

Further, **\*\*CivicScope doesn't consider race or ethnicity of individuals in our tool.\*\*** This is not to say that we aren't using variables that might be closely correlated with race and ethnicity. We use weather and historical violent crime data to run our risk scores. We do include a geographic component, a cell area. While this does not contain race or income information directly — intentionally — we acknowledge that in some cases, location of a crime event can include information that is indirectly related to race, ethnicity and income.

\n, [dkg 2017, line 784], evaluation\_notebooks/notebooks/PreventingBias.ipynb

Such open self-conscious statements in code made publicly available seem to raise CivicScope beyond the kinds of reproach directed at other software. However, this last apology suggests that they are also aware of foundational flaws in such a method. A system whose code does not “consider race” has disproportionate effects on different racial groups in the United States because racial identity intersects with socio-economic status and with urban neighborhoods of neglect thanks to absentee landlords and reduced funds for civic beautification. This statement therefore admits that race is indeed a part of the code of predictive policing even when it is not explicitly present in obvious tokens and

36

ontologies.

This reading of street trees, therefore, is not to suggest that CivicScape or other predictive policing can operate ethically or “get it right” if only they utilize the correct inputs. However, the code and notebooks of CivicScape attempt to present that argument as they invite us to examine their code. What the CivicScape example makes clear is that the big data approach to policing relies primarily on the premise of uncritical correlations that reify existing divisions between how people and neighborhoods are treated rather than offering a fair and just machine learning solution. As a demonstration piece, CivicScape seems to offer not so much crime prevention software as a confirmation bias engine. Furthermore, this case study highlights the limits of reading demonstration versions of software. Surely, we do not have access to the great catalogs of data police precincts input about crime in their districts. We have no way of knowing their contents, which essentially equals legislated policing without oversight. However, the street trees input suggests that such inputs may be used blindly and somewhat arbitrarily to produce results justified by their alignment with already-held beliefs.

37

In this analysis we are using a showpiece version of the software with sample training information. Presumably, once implemented, the model is adapted and tweaked to variables more attuned to the particular neighborhood. However, our initial foray into this code is sufficient to consider some overarching questions. Even if we imagine a much smarter system, one with complete models of all of the complaints and a perfect weighting system, a larger logical dilemma emerges. Principally, is not predictive policing software in essence a negative feedback loop?

38

Consider the case of the neighborhood identified by the software as an area in need of policing. Taking into consideration the recommendations of this and other software, the police decide to dispatch more officers to that neighborhood. More officers equals more eyes on the ground who can then observe more crime. Now, the neighborhood warrants even more policing. Or consider an even more concerning case, given the documented accounts of police violence, especially against BIPOC and LGBTQ+ citizens: what happens when the police are the cause of the crimes? Consider the accounts of police violence at the 2020 Queer Liberation March for Black Lives and Against Police Brutality in New York [Walker 2020] or at Pride events the following year [Hart 2021]. The software does not take any acts of aggression by the police into account, nor does it account for the disproportionate dispatch of police to an area due to conditions associated with poverty. The software also fails to account for other forms of crime being perpetrated in neighborhoods, such as white collar crime and rent manipulations that force people out of their homes.

39

Ultimately, the software alone offers only one piece of the decision-making process. However, its lines of code open avenues for further investigation into how policing decisions are being made. Keeping this information from the citizens who are being governed by them is an act of state control via obfuscation, which changes the legislative system into something unknowable and mysterious, that which cannot be overseen and made clear to the public. Even assuming the best possible intentions for those doing the policing, to accept the premise that machine learning is beyond supervision is to surrender even more authority to the State just at the moment that the decision-making process could be made transparent in a way never before possible.

40

## Access to Show Code versus Source Code?

Unlike other software for predictive policing, the code for CivicScape is available for review on Github. However, to say that the code for CivicScape is open access is not quite accurate. What the company has released is a version of the software that functions as a kind of demo, without any of the specific data or customizations that particular police departments would use. As code readers, then, we realize we have entered into a bit of a Potemkin Village. The above reading is a commentary on what is available to speak about one implementation of this software, but it must provide the basis for further analysis and critique.

41

When it comes to predictive policing software, transparency is voluntary. Contrary to analogous laws in the United Kingdom, the Freedom of Information Act (FOIA) in the United States allows exemptions which protect software used in police departments. When a person submits an FOIA request for a software's source code or algorithm, the input data, or the police's use of that software and its predictions, the receiving law enforcement agency can withhold that information if they provide reasons pursuant to a list of federal exemptions. There are three paths that such agencies can use to block these requests. For one, the courts have a long history of denying FOIA requests for the intellectual

42

property of government contractors — trade secrets that, if made public, could potentially harm the commercial interests of the software provider [Bakke 2018].<sup>[11]</sup> This “business information exemption” is usually enough to deny a request for the source code or algorithm. But the FOIA also includes a robust exemption for requests related to law enforcement. In this case, Erik Bakke highlights two clauses: 7C and 7E. The first protects the personal privacy of individuals, notably including any input data from geographical surveillance considered residential. The second keeps confidential any strategies for law enforcement that might be compromised if revealed to its target suspects, which covers *how* the police use the software. Because legal precedent has extended this latter clause to anyone suspected of future crime, predictive algorithms — which define their own theoretical suspects — have received near blanket immunity.<sup>[12]</sup>

To state the obvious: the courts are invested in the advancement of law enforcement. For this reason, law enforcement agencies have had little trouble meeting the burden of proof for the above exemptions. Even in cases where they have agreed to comply, such agencies have gotten away with responding to FOIA requests only after the information is out of date, or so heavily redacted or disorganized so as to be useless [Bakke 2018, 168]. While some states tend to be more pro-access than others — with Florida, Ohio, and Vermont allowing the most transparency, and Pennsylvania and Washington D.C. the least — every state’s open records, public records, or “Sunshine” laws are modeled after the federal act and, when challenged in court, have historically fallen back on the federal exemptions list as the guiding legal precedent [Bakke 2018, 157].<sup>[13]</sup> Functionally, then, as Bakke concludes, “the current legal framework provides little opportunity for substantial transparency with predictive policing” [Bakke 2018, 170]. Thus, when services like CivicScope publish their source code, that code does not reflect input data not already public (such as census tracts, published crime statistics, and 311 reports), nor the exact methods by which these algorithms deliver predictions to the police. Even if the company behind predictive policing software wanted to commit fully to transparency, the above legal protections for law enforcement agencies, as well as civil rights concerns regarding surveillance data, would ultimately secure crucial segments of information related to predictive policing behind blue walls and black boxes.

43

Furthermore, the term “open source” is a misnomer here, misrepresenting the predictive policing system as an open, transparent, neutral system. Instead, its technical, financial, and knowledge barriers continue to prevent access to the “accessible” code. CivicScope’s Github repository showcases a non-deployed version of its source code, without any of the customization implemented for its police department clients. It is neither the exact code they used to develop their tool nor the production-ready code their customers use. It was not developed with an open-source ethos or methodology, which would trace the authorship and changes over time during its development using version control so that the public could see how decisions were made. Rather, each file was uploaded to the repository in its final version without any changes tracked. Second, it does not include any of the input data necessary to run the CivicScope tool, neither as processed by their tool nor even linking to any original sources. Their “DataInputsPractices” notebook outlines the detailed and varied formatting that must occur to turn each aggregated crime report into a viable dataset for input, as well as how these differ from city to city and by type of crime and collection method. After some consternation, we were able to open their Jupyter notebooks, albeit with troubleshooting that required slight modifications.<sup>[14]</sup> Yet without any sample data from CivicScope and without data files for input, the available code literally could not be run. Users, including us, are left to find or create their own data to process, which will fundamentally impact and skew the outputs and their findings. Because of this, CivicScope’s public release acts as “show code”, offering a sample of what its predictive policing code is *like* rather than what it *is*. Third, even if users were able to reproduce synthetic data or locate a viable dataset, running such code requires costly compute power, setup time, and expertise.<sup>[15]</sup>

44

Similar to overwhelming an opposing counsel with truckloads of paperwork under the guise of “full disclosure”, simply dumping the (in)complete code on Github is not enough to make it usable or understandable. This is what Mike Ananny and Kate Crawford call a “resistant transparency” [Ananny and Crawford 2018]. In this case, portions can be read and analyzed, but the instructions for implementing the code as part of a complete, functional object require financial outputs and significant technical expertise. Providing what is effectively partial openness to a software system means that it remains too complex to be analyzed and understood by outside observers. This amounts to performative transparency.

45

CivicScope’s goal to move toward “open and transparent [predictive policing], to enable trust and encourage constructive scrutiny” is notable. However, rather than reveal a less biased approach to predictive policing, if that is

46

even possible, instead it reveals and continues to uphold the flawed assumptions at the heart of this industry's algorithmic methodologies. While preferable to proprietary black boxes, open-sourcing alone does not address the larger questions that remain around which crimes should be predicted, how, and what it means to do so. It also suggests further questions at stake for algorithmic transparency and open source in general. Ananny and Crawford argue that “making one part of an algorithmic system visible — such as the algorithm, or even the underlying data — is not the same as holding the assemblage accountable” [Ananny and Crawford 2018, 984]. CivicScape's well-intentioned effort demonstrates that an open-source ethos toward algorithmic transparency may also need to expand to consider usability and community feedback in order to address the high stakes of predictive policing. Performative transparency should not be the smoke screen that prevents a more systems-based approach to algorithmic accountability.

## Conclusion

Citizens deserve access to the code that governs their lives. Software, like the predictive policing program discussed in this essay, are algorithmic processes enacted on citizens that shape and control their lives. Therefore, along with Lawrence Lessig, we acknowledge that code is law [Lessig 2006, 110]. If an aberrant tree root in your sidewalk is causing your community to be assigned a higher risk score and to be classified as a “bad” neighborhood, you deserve to know. If the software designed to classify “bad” neighborhoods leads to more policing and more arrests of BIPOC community members, that cultural logic must be interrogated. And why is increased policing the goal, when perhaps the system should be allocating landscapers (and by analogy, electricians or, more importantly, mental healthcare workers)? Furthermore, if your reports to 311 centers for overgrown tree roots are being used to determine the crime risk score of your neighborhood, even indirectly by providing training data for machine learning systems, you should be informed when you call into the center or access the app. However, a mere recorded warning or pop-up message to dismiss will not suffice. Community members need to have a critical understanding of how their data is being collected and used. Public awareness campaigns, community town halls, and courses on civics should include this new, though mostly hidden, aspect of governance. The foundational history lesson of “no taxation without representation” becomes the call “no data collection without open inspection”.

47

Currently, artificial intelligence software is shielded through obfuscation, rendering police methods and procedures inscrutable behind a veil of unknowability. Even beyond the realm of predictive policing, artificial intelligence software is cordoned off as unknowable or unexplainable. Recently, scholars such as Huah and Raley and Berry have been developing approaches to examine this part of the software. While the so-called “explainability problem” is based on a technical description of inaccessible processes, there is plenty of accessible code in machine learning systems. If we are going to interrogate software, particularly as it is used in a governance capacity, such as predictive policing or bail-setting, we cannot stop at the police caution tape around the source code. While examining inputs and outputs gives a sense of what the software *does*, without the complete code such analysis offers little insight into *how* the processes and rules lead to those outcomes. The situation is analogous to noticing the disproportionate numbers of jailed black and brown bodies in America without gaining a sense of how that happened. Or, to put it more accurately, to study the inputs and outputs without examining the code allows us to observe only the conditions and effects of injustice without tracing out the inequitable procedures and methods that lead to it. At the same time, we need to acknowledge that the code that is released to the public may be acting as a kind of Potemkin Village, screening us from the implemented code.

48

In our reading of CivicScape, we attempt to open up the code to a variety of readings and critiques of that power, including issues of social control, policing, and ecological concerns. In other words, once we can move beyond narrow investigations of software use cases, reported data and results (that is, a black-boxed understanding of how the system works in a particular context), we can narrow the range of the unexplainable by analyzing the interpretable, interrogating the underlying assumptions and power relations encoded in the software, including factors that were altogether unexpected, such as the critical role of street trees. Like their literal biological counterpart, the roots of these street trees have broken through a surface understanding of the software, disrupting the smooth cement of an imaginary, algorithmically engineered city, one where crime is tidy, predictable, and unprejudiced.

49

Our investigation of the source code for this predictive policing software suggests the need for a new “Sunshine”

50

movement, one that demands that the code for all software operating on citizens be made accessible to all who are governed by it. As part of the Humanities and Critical Code Studies Lab, we have created such a project, the Sunshine Source Force, in alliance with the Algorithmic Justice League and those working in the spheres of Critical AI and Data Justice, such as the Data Sitters Club. We call for governments to open their code to the public and to make the accessible the operating systems of government. To open AI systems, officials should make available the training data and weights. To be truly legible, they should clearly document and explain the code in terms that non-specialists can understand. A law enforcement agency might ask whether releasing the code would give away valuable information to those it is trying to police, similar to an armed force revealing strategic military data in the middle of a war. To that we can only say that a police force should not be at war with those who have authorized them to preserve and protect their community. If code is law, then we who are governed by it should by law have access to the code. Governmental actors should not be allowed to hide behind the mythical mystery of machine learning.

## Notes

[1] Algorithmic injustice has been increasingly flagged by scholars like Ruha Benjamin, Safiya Noble, and Joy Buolamwini and her Algorithmic Justice League. In January 2021, we launched the Anti-Racist Critical Code Studies Reading Group, sponsored by the Humanities and Critical Code Studies Lab (USC), Creative Code Collective (USC), the Digital Arts and Humanities Commons (UCSB), the Digital Humanities Initiative at SDSU, and Feminist.AI.

[2] Throughout this essay, we use various terms to describe citizens who are not white, depending on the rhetorical context. We use “black and brown bodies” in line with activists such as bell hooks and Angela Davis.

[3] Outside of the United States, the Chinese state has employed predictive policing to persecute the Uyghurs, and a predictive policing system called PRE-COBS, is popularly used in central Europe.

[4] *Moneyball* is the title of a 2003 book by Michael Lewis (and 2011 film adaptation) that has become synonymous with the use of statistical analysis to improve the winning prospects of a major league baseball team (and other enterprises).

[5] CivicScape was also shaped by Anne Milgram, former New Jersey Attorney General and chair of the board of directors.

[6] “Explainability” is the ability of humans to explain the unseen associations resulting from machine learning processes. “Interpretability” is the ability of humans to explain the explicit or specified connections based on known causes and effects in the system. The interpretability problem is the challenge of ever knowing what happens in the process of machine learning. See [Johnson 2020] for more on this distinction. For a fascinating approach to problems of interpretability that contests the notion of explainability using creative coding, we recommend Catherine Griffiths’ *Toward Counteralgorithms* (2021).

[7] See

[https://github.com/dkg/CivicScape/blob/2059d278fbed162c0c174611bbc4fec83180495/evaluation\\_notebooks/notebooks/DataInputsPractices.ipynb](https://github.com/dkg/CivicScape/blob/2059d278fbed162c0c174611bbc4fec83180495/evaluation_notebooks/notebooks/DataInputsPractices.ipynb).

[8] See [https://github.com/dkg/CivicScape/blob/2059d278fbed162c0c174611bbc4fec83180495/Training\\_and\\_Testing/02.Code/philadelphia\\_pa/40.create\\_311.sql](https://github.com/dkg/CivicScape/blob/2059d278fbed162c0c174611bbc4fec83180495/Training_and_Testing/02.Code/philadelphia_pa/40.create_311.sql).

[9] Again, we note that this released code is for journalistic review and does not reveal the modified versions of the code as it is implemented, but we do this reading as a “good faith” exercise in reviewing what is available (as the gesture of an open-source predictive policing algorithm suggests) to indicate the kinds of readings we could perform if the implemented code were made public on the grounds of transparency, which we argue it should be.

[10] Street trees are admittedly a kind of input into this system. However, since they are part of the training data, we would not have known of their role in the formation of this system without looking at the lines of code.

[11] According to Bakke, the 4th exemption covers “privileged or confidential [...] trade secrets and commercial or financial information” [Bakke 2018]. Bakke uses the example of a denied FOIA request for the blueprints of government-contracted voting machines.

[12] The two exemptions read as follows: “7(C). Could reasonably be expected to constitute an unwarranted invasion of personal privacy”; and “7(E). Would disclose techniques and procedures for law enforcement investigations or [...] disclose guidelines for law enforcement investigations [...] if such disclosure could reasonably be expected to risk circumvention of the law”. Bakke cites the legal case of American Civil

Liberties Union of New Jersey v. FBI, in which an FBI racial mapping initiative was exempted because disclosing the distribution of future surveillance would interfere with later enforcement proceedings and reveal the targets of those efforts.

[13] Prior to 2008, the Pennsylvania Right to Know Act was regarded as one of the worst in the country. The pre-2008 law placed the burden of proof on the person filing the request. The new law states that all documents will be presumed to be open to the public unless the agency holding them can prove otherwise.

[14] Source was cloned from <https://github.com/dkg/CivicScape.git> and staged to an online, shared CoLab Notebook. More information available at: <https://colab.research.google.com/drive/1fGZ1vtk9iy6N1QCBNAT1jA2Ey5cLTdyz?usp=sharing>.

[15] A Reddit user nick898 (2017) on r/DataScience, which is a forum for “practitioners and professionals”, sums up the problem: “The readme in the Training and Testing folder explains how to run the model, however it’s not clear to me how to set up the system myself. They list a number of requirements which I can go and download myself and ... they assume you’re using a [sic] Amazon Web Services (AWS) account with the ‘appropriate AWS instances’ that are required for their code. I’d like to try running their code, but it’s not clear to me how to get it all set up. I’ve never used AWS before and it’s not clear to me exactly what I do to run their model on my own”.

## Works Cited

- Ananny and Crawford 2018** Ananny, M. and Crawford, C. (2018) “Seeing without knowing: Limitations of the transparency ideal and its application to algorithmic accountability”, *New Media & Society*, 20(3), pp. 973–989. <https://doi.org/10.1177/1461444816676645>.
- Angwin, Kirchner, and Surya 2016** Angwin, J. et al. (2016) “Machine bias”, *ProPublica*, 24 February. <https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing> (Accessed: 24 February 2022)
- Bakke 2018** Bakke, E. (2018) “Predictive policing: The argument for public transparency”, *Annual Survey of American Law*, 74(1). Available at: <https://annualsurveyofamericanlaw.org/wp-content/uploads/2019/08/74-1-Predictive-Policing-The-Argument-for-Public-Transparency.pdf>.
- Benbouzid 2019** Benbouzid, B. (2019) “To predict and to manage: Predictive policing in the United States”, *Big Data & Society*, 6(1). <https://doi.org/10.1177/2053951719861703>.
- Benjamin 2019** Benjamin, R. (2019) *Race after technology: Abolitionist tools for the New Jim Code*. Cambridge, UK: Polity.
- Berry 2023** Berry, D. M. (2023) “Tracing toxicity through code: Towards a method of explainability and interpretability in software”, *Digital Humanities Quarterly*, 17(2). Available at: <https://www.digitalhumanities.org/dhq/vol/17/2/000706/000706.html>.
- Brustein 2017** Brustein, J. (2017) “The ex-cop at the center of controversy over crime prediction tech”, *Bloomberg Quint*, 10 July. Available at: <https://www.bloombergquint.com/markets/the-ex-cop-at-the-center-of-controversy-over-crime-prediction-tech>.
- Burley 2018** Burley, B.A. (2018) “Green infrastructure and violence: Do new street trees mitigate violence crime?”, *Health Place*, 54, pp. 43-39. <https://doi.org/10.1016/j.healthplace.2018.08.015>.
- CodeFest 2021** CodeFest (202) *Critical code studies*. 2 October. Available at: [https://www.youtube.com/watch?v=\\_oKflqNPMXA](https://www.youtube.com/watch?v=_oKflqNPMXA).
- Coleman 2009** Coleman, B. (2009) “Race as technology”, *Camera obscura: Feminism, culture, and media studies*, 24(1), pp. 177–207. <https://doi.org/10.1215/02705346-2008-018>.
- Dixon-Roman, Nyame-Mensah, and Russell 2019** Dixon-Roman, E., Nyame-Mensah, A., and Russell, A. R. (2019) “Algorithmic legal reasoning as racializing assemblages”, *Computational Culture*, 7. Available at: <http://computationalculture.net/algorithmic-legal-reasoning-as-racializing-assemblages/>.
- Donovan and Prestemon 2010** Donovan, G.H. and Prestemon, J.P. (2012) “The effect of trees on crime in Portland, Oregon”, *Environment and Behavior*, 44, pp. 3-30. <https://doi.org/10.1177/0013916510383238>.
- Egbert and Krasmanb 2020** Egbert, S. and Leese, M. (2021) “Criminal futures: Predictive policing and everyday police work”. London: Routledge.
- Egbert and Leese 2021** Egbert, S. and Krasmanb, S. (2020) “Predictive policing: Not yet, but soon preemptive?”, *Policing and Society*, 30(8), pp. 905–919.
- Fried et al. 2020** Fried, G., et al. “Assessing CivicScape: The value of applying open source approaches toward algorithmic

decision system accountability”, *AI@WORK 2020*. Amsterdam, Netherlands, 5-6 March. Available at: <https://reshapingwork.net/ai/session/assessing-civicscape-the-value-of-applying-open-source-approaches-toward-algorithmic-decision-system-accountability/>.

- Griffiths 2021** Griffiths, C. (2021) *Towards counteralgorithms: The contestation of interpretability in machine learning*. PhD thesis. University of Southern California.
- Hart 2021** Hart, R. (2021) “Police clash with crowds at New York Pride on first year NYPD is barred from event”. *Forbes*, 28 June. Available at: <https://www.forbes.com/sites/roberthart/2021/06/28/police-clash-with-crowds-at-new-york-pride-on-first-year-nypd-is-barred-from-event/?sh=753bd9b87ef2>.
- Heaven 2020** Heaven, W. D. (2020) “Predictive policing algorithms are racist. They need to be dismantled”, *MIT Technology Review*, 17 July. Available at: <https://www.technologyreview.com/2020/07/17/1005396/predictive-policing-algorithms-racist-dismantled-machine-learning-bias-criminal-justice/>.
- Hong 2015** Hong, S. (2015) “Subjunctive and interpassive ‘knowing’ in the surveillance society”, *Media and Communication*, 3(2), pp. 63-76.
- Hong and Szpunar 2019** Hong, S. and Szpunar, P. M. (2019) “The futures of anticipatory reason: Contingency and speculation in the sting operation”, *Security Dialogue*, 50(4), pp. 314–330.
- Hua and Raley 2023** Hua, M. and Raley, R. (2023) “How to do things with deep learning code”, *Digital Humanities Quarterly*, 17(2). <https://www.digitalhumanities.org/dhq/vol/17/2/000684/000684.html>.
- Human Rights Watch 2018** Human Rights Watch (2018) *“Eradicating ideological viruses”: China’s campaign of repression against Xinjiang’s Muslims*. Available at: <https://www.hrw.org/report/2018/09/09/eradicating-ideological-viruses/chinas-campaign-repression-against-xinjiangs>.
- Johnson 2020** Johnson, J. (2020) *Interpretability vs. explainability: The black box of machine learning*, *BMC Blogs*, 16 July. Available at: <https://www.bmc.com/blogs/machine-learning-interpretability-vs-explainability/> (Accessed: 9 February 2022).
- Kaufmann, Egbert, and Leese 2019** Kaufmann, M., Egbert, S., and Leese, M. (2019) “Predictive policing and the politics of patterns”. *British Journal of Criminology*, 59(3), pp. 674–692.
- Kelling and Wilson 1982** Kelling, J.Q. and Wilson, G.L. (1982) *Broken windows*, *The Atlantic*, March. Available at: <https://www.theatlantic.com/magazine/archive/1982/03/broken-windows/304465/> (Accessed 5 October 2021).
- Kendi 2019** Kendi, I. X. (2019) *How to be an antiracist*. New York: One World.
- Lally 2021** Lally, N. (2021) “‘It makes almost no difference which algorithm you use’: On the modularity of predictive policing”, *Urban Geography*, 43(9), pp. 1437-1455.
- Lessig 2006** Lessig, L. (2006) *Code: And other laws of cyberspace, Version 2.0*. New York: Basic Books.
- Lum and Isaac 2016** Lum, K. and Isaac, W. (2016) “To predict and serve?” *Significance*, 13(5), pp. 14–19. <https://doi.org/10.1111/j.1740-9713.2016.00960.x>.
- Marino 2006** Marino, M. C. (2006) “Critical code studies”, *electronic book review*, 4 December. Available at: <https://electronicbookreview.com/essay/critical-code-studies/>.
- Marino 2020** Marino, M. C. (2020) *Critical code studies*. Cambridge, MA: The MIT Press.
- McPherson 2013** McPherson, T. (2013) *U.S. operating systems at mid-century: The intertwining of race and UNIX: Race after the internet*. London: Routledge.
- Meijer and Wessels 2019** Meijer, A. and Wessels, M. (2019) “Predictive policing: Review of benefits and drawbacks”, *International Journal Of Public Administration*, 42(12), pp. 1031–1039.
- Noone 2021** Noone, G. (2021) “The case against predictive policing”, *Tech Monitor*, 1 July. Available at: <https://techmonitor.ai/technology/ai-and-automation/case-against-predictive-policing>.
- O'Donnell 2019** O'Donnell, R. M. (2019) “Challenging racist predictive policing algorithms under the Equal Protection Clause notes”, *New York University Law Review*, 94(3). Available at: <https://www.nyulawreview.org/wp-content/uploads/2019/06/NYULawReview-94-3-ODonnell.pdf>.
- Polack 2020** Polack, P. (2020) “Beyond algorithmic reformism: Forward engineering the designs of algorithmic systems”, *Big Data & Society*, 7(1). <https://doi.org/10.1177/2053951720913064>.

**Richardson, Schultz, and Crawford 2019** Richardson, R., Schultz, J., and Crawford, K. (2019) *Dirty data, bad predictions: How civil rights violations impact police data, predictive policing systems, and justice*. Available at: <https://papers.ssrn.com/abstract=3333423>.

**Sandhu and Fussey 2021** Sandhu, A. and Fussey, P. (2021) “‘The ‘Uberization of policing’? How police negotiate and operationalise predictive policing technology”, *Policing and Society*, 31(1), pp. 66–81.

**Sankin et al. 2021** Sankin, A. et al. (2021) “Crime prediction software promised to be free of biases. New data shows it perpetuates them”, *The Markup*, 2 December. Available at: <https://themarkup.org/prediction-bias/2021/12/02/crime-prediction-software-promised-to-be-free-of-biases-new-data-shows-it-perpetuates-them>.

**Schellenberg 1997** Schellenberg, K. (1997) “Police information systems, information practices and individual privacy”, *Canadian Public Policy / Analyse de Politiques*, 23(1), pp. 23–39. <https://doi.org/10.2307/3552129>.

**Sheehey 2019** Sheehey, B. (2019) “Algorithmic paranoia: The temporal governmentality of predictive policing”, *Ethics and Information Technology*, 21, pp. 49–58.

**Walker 2020** Walker, J. W. (2020) “51 years after Stonewall, New York’s queer liberation march faces police violence”, *The Nation*, 3 July. Available at: <https://www.thenation.com/article/society/queer-liberation-march-police-violence/>.

**Wood 2019** Wood, S. E. (2019) “Policing through platform”, *Computational Culture*, 7. Available at: <http://computationalculture.net/policing-through-platform/>.

**Yuill 2019** Yuill, S. (2019) “Critical approaches to computational law”, *Computational Culture*, 7. Available at: <http://computationalculture.net/section-editorial-critical-approaches-to-computational-law/>.

**dkg 2017** dkg (2017) *Understanding the CivicScape data and model*. Available at: <https://github.com/dkg/CivicScape>.

**nick898 2017** nick898 (2017) “Predictive policing model released by CivicScape”, *Reddit*, 27 March. Available at: [https://www.reddit.com/r/datascience/comments/61p9tx/predictive\\_policing\\_model\\_released\\_by\\_civicscape/](https://www.reddit.com/r/datascience/comments/61p9tx/predictive_policing_model_released_by_civicscape/).



This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.