Exposing Error in Poverty Management Technology

A Method for Auditing Government Benefits Screening Tools

NEL ESCHER and NIKOLA BANOVIC, University of Michigan, USA

Public benefits programs help people afford necessities like food, housing, and healthcare. In the US, such programs are means-tested: applicants must complete long forms to prove financial distress before receiving aid. Online benefits screening tools provide a gloss of such forms, advising households about their eligibility prior to completing full applications. If incorrectly implemented, screening tools may discourage qualified households from applying for benefits. Unfortunately, errors in screening tools are difficult to detect because they surface one at a time and difficult to contest because unofficial determinations do not generate a paper trail. We introduce a method for auditing such tools in four steps: 1) generate test households, 2) automatically populate screening questions with household information and retrieve determinations, 3) translate eligibility guidelines into computer code to generate ground truth determinations, and 4) identify conflicting determinations to detect errors. We illustrated our method on a real screening tool with households modeled from census data. Our method exposed major errors with corresponding examples to reproduce them. Our work provides a necessary corrective to an already arduous benefits application process.

CCS Concepts: • **Human-centered computing → Human computer interaction (HCI)**; • **Applied computing → Law**.

Additional Key Words and Phrases: e-government; algorithmic audit; automated decision systems

ACM Reference Format:
Nel Escher and Nikola Banovic. 2020. Exposing Error in Poverty Management Technology: A Method for Auditing Government Benefits Screening Tools. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW1, Article 64 (May 2020), 20 pages. https://doi.org/10.1145/3392874

1 INTRODUCTION

Public benefits programs in the United States help low-income people afford food, housing, healthcare, and other necessities. Such programs are means-tested: welfare applicants must prove that they are in a dire financial situation before receiving assistance. The official benefits applications often stretch dozens of pages, soliciting personal information about household members and minutia on their employers, assets, and expenses. If applicants provide a wrong value for certain questions, it could lead to criminal fraud charges. With such complexity and stringent demands for accuracy, it is unsurprising that millions of people who are eligible for benefits do not receive them [35].

Eligibility screening tools that are offered online can provide a helpful abridgement of the rules. They offer potential applicants a short, accessible form that accepts estimated information and predicts household eligibility. Such tools issue predictions automatically, as opposed to systems that support human decisions (e.g., online benefits tools that collect information from clients, but then pass the information to an eligibility technician for further processing [51]). They could reduce

Authors' address: Nel Escher, kescher@umich.edu; Nikola Banovic, nbanovic@umich.edu, University of Michigan, Ann Arbor, Michigan, USA, 48109.

2573-0142/2020/5-ART64 $15.00
https://doi.org/10.1145/3392874

the burden on the legal aid organizations that provide counseling to people who have difficulty navigating the public benefits system.

While these tools are designed to help struggling, overwhelmed people decide whether to embark on the lengthy benefits application process, their inaccurate eligibility predictions may prevent deserving households from seeking benefits. Inaccurate predictions are dangerous because people tend to rely on information they discover on government services websites, even if it is wrong [26].

Though online administration can improve efficiency and reduce the demands on government workers, technology used for poverty management can be sloppily implemented and prone to error [21]. New legislation or changes in benefits manuals could make such tools out of date, even if temporarily. Frequent budgetary cuts to benefits programs keep the departments responsible for implementing and maintaining those tools short-staffed [43, 51]. Different government agencies either develop these tools in-house or contract them out to the private sector. Despite all the debugging and quality assurance methods that software engineers have in their toolbox, online benefits administration systems are still incorrectly implemented [17].

Errors can be exposed on a case-by-case, household-by-household basis when savvy applicants contact benefits lawyers about suspicious determinations. Yet, posing a well-founded challenge to a screening tool prediction requires knowledge of the complete guidelines; it is unrealistic to expect potential applicants to possess the very knowledge they seek from the tool. Even if error is suspected, each household provides only one example, so the extent of problems is not immediately discoverable. Unlike full applications, screening tools do not produce an official determination and so do not generate a paper trail. Tools that offer promises of confidentiality will erase any information entered about a household after returning their results (e.g., [4]). This leaves no opportunity to find out if potential applicants have actually received erroneous determinations and prevents investigating the performance of the screening tool on real inputs.

The code for existing screening tools is typically not open for inspection, so they function as algorithmic decision-making black-box systems. One way to probe such systems is with an algorithmic audit [14, 27, 44, 46, 50]. In particular, scraping audits [46] are useful for testing online systems when source code is unavailable; they involve submitting multiple queries to a system and recording the results. However, systems audited in the past do not have a direct measure of correctness (e.g., there is not an objective list of search results that ought to be returned in response to a Google query [50]), so past auditing approaches have not described how to specify the full, correct behavior for algorithmic decision-making systems. To assess screening tools for correctness, their predictions must be compared against the behavior prescribed by law.

Our primary contribution is methodological: we present a method that allows for immediate audit of the correctness of online benefits screening tools on a large set of realistic inputs. We focus on bright-line rule-based requirements for the receipt of benefits, which only permit a single interpretation and allow us to produce a formal model that exactly encodes the eligibility guidelines from statutes and policy manuals [32]. Unlike automated software tests [42] that randomly generate many test samples to test as many program branches as possible or search for the smallest set of inputs that test all branches of a program, we focus on generating realistic inputs that allow us to estimate the effect of screening tool errors on real people.

Our method first generates a set of realistic test households using data and models from surveys representative of the population covered by the screening tool's jurisdiction. Then, our automated data entry scripts populate the target screening tool questions using the test household information. We match screening tool determinations for each household against determinations from our eligibility guidelines formalization to identify incorrect determinations. We explore the wrong determinations to diagnose specific errors and understand who those errors impact the most.

We illustrated and evaluated our method on the Pennsylvania “Do I Qualify?” website [4] and tested households that we generated using models based on US census data. The “Do I Qualify?” tool provides eligibility predictions for five different public benefits: healthcare, food assistance, cash assistance, free or reduced price school meals, and subsidized child care. We targeted this particular tool because the complexity of its interface makes it one of the most challenging to audit. We used Selenium [8], a web browser automation tool, to automatically enter information into the screening tool and recorded the determinations that the screening tool returned for each test household. We inspected those determinations by comparing them to our formalization of the Pennsylvania benefits rules, which should inform the screening tool’s determination logic.

Our evaluation identified major errors with the Pennsylvania “Do I Qualify?” screening tool and generated a set of realistic test households to reproduce the errors. We found that the tool strongly departed from our statutory guidelines formalization. As an example, for the subsidized child care benefit, it predicted every test household was ineligible, though at least 4.6% of our test households were potentially eligible for the benefit. For all benefits we tested, we were able to identify provisions from patterns of error that the screening tool did not correctly implement.

Our work contributes a quantitative measure for embedded error in a subset of systems that provide government services (i.e., “electronic government,” or e-government). Our method produced a dataset that allowed us to discover suspicious determinations without requiring real applicants to use the tool. It quantified the performance of a real screening tool on realistic inputs and generated a paper trail with a list of examples to contest incorrect determinations. Our method gives private citizens and nonprofit organizations an auditing approach with which to apply additional pressure on the government agencies that provide broken screening tools, so that people who rightly qualify for benefits are encouraged to apply.

2 AUTOMATING PUBLIC BENEFITS ADMINISTRATION

Securing poverty relief in the United States can be a long process. Applicants may face psychological barriers and technical hurdles. Means-testing is associated with a surveillance regime, where the process of determining eligibility requires the extraction of income and asset data [20]. Applicants must provide evidence that their personal financial situation causes them substantial hardship. The format of the required proof is standardized in statutes and lengthy benefits manuals [38–40].

Each benefit has its own set of guidelines, and some benefits are offered through various programs pitched at different applicant pools (e.g., medical assistance is available through Medicare, the Children’s Health Insurance Program, Former Foster Care Youth provisions, and dozens of other categories [39]). Requirements can variously set income limits, resource limits, and group composition rules, as well as impose obligations such as job skills training. The information provided by the applicants must be verified and assessed by government agencies to determine eligibility.

Receiving public benefits is often stigmatized. The 2018 General Social Survey [48] polled Americans on their level of agreement with the following statement: “The social benefits from the government make people lazy.” Over half (50.9%) of the respondents agreed or strongly agreed [48]. Political rhetoric presenting welfare recipients as irresponsible or addicts contributes to the narrative that these programs transfer money from taxpayers to undeserving people, and helps justify the attachment of punitive, invasive conditions to the receipt of public benefits [33].

Moving the benefits application process online has the potential to reduce stigma [33], increase access [51], and improve the speed and consistency of eligibility decisions [31]. When benefits administrators assess applications, they may reproduce stigma when interacting with their clients and their unconscious bias can affect their evaluation of benefits applications [33]. Automated verification thus reduces the role that administrator discretion plays in determining eligibility. For paper applications accepted through mail, applicants must acquire envelopes and stamps; use of an online portal can reduce barriers to submission [31] (e.g., interviews with individuals experiencing homelessness showed that public computers at libraries offer access opportunities at least to those who are literate [29]). Although the benefits of transition to e-government do not accrue to all, it can still improve the public services experience for some users.

Screening tools are part of the broader project of e-government; they provide instant advice about eligibility that could ameliorate the benefits application process. Applicants officially find out whether they are eligible to receive benefits after their application is processed, supporting documents are verified, and investigative interviews are completed. Screening tools ease this wait; they offer potential applicants a quick form that solicits household information and applies some of the same tests used by the automated decision-making systems that evaluate full applications. Some screening tools prioritize user convenience and include few questions (e.g., the Massachusetts Department of Transitional Assistance provides a tool which promises potential applicants a SNAP eligibility prediction in 10 seconds [5]), while others incorporate more questions to better capture the complexity of the eligibility guidelines. No matter which values direct development, it is arguably better to be over-inclusive when predicting eligibility. Filling out superfluous forms can be aggravating, but that cost is low compared to losing access to medical care, housing, or food.

However, existing automated decision-making systems used for public benefits administration have an uneven track record. For example, a past attempt by IBM to automate the Indiana state welfare system incorrectly rejected thousands of applicants, limiting or removing their access to healthcare and other services [25]. In some cases, the applicants died while their appeal was pending [21]. The system implementation was so poor that the Indiana Supreme Court ruled IBM had materially breached the services agreement [17]. The system remained in place for three years; it was only cancelled after significant public outcry brought attention to the thousands of people wrongly denied benefits [25].

The issuance of incorrect determinations is difficult to contest, if error is even suspected, since the applicants are usually very vulnerable and have limited political power. The MiDAS system in Michigan issued more than 34,000 erroneous unemployment fraud determinations and wrongly extracted payment from out-of-work Michiganders [13]. It took the Michigan Unemployment Insurance Agency two years to correct and address the issue. When automated benefits technology produces inaccurate results, broken systems can remain in place for years because those automated benefits systems often operate with limited or no human oversight [13].

Screening tools can seem equally authoritative, so potential applicants may act based on their predictions without challenging them. Although some people who have low trust in the government are also skeptical about decision-making systems that administer public services [11], others have been shown to over-trust technology that claims authority, even when they have reason to doubt its decisions [45]. Screening tools are often provided by the same agencies that administer full applications, and links to the screening tool can appear side-by-side with links for officially applying (e.g., the Pennsylvania COMPASS "Do I Qualify?" screening tool [4]). If a screening tool returns incorrect results, perceptions of unfairness may be justified. Erroneous predictions that masquerade as reliable advice could shatter user expectations and trust, undermining the effectiveness of e-government initiatives [24].

A growing body of research [14, 44, 46, 50] uses algorithmic auditing to probe the behavior of black-box algorithmic systems, some of which are similar to screening tools. Screening tools accept information from users, process the input information according to some internal, opaque algorithm, and return a decision. The traditional audit study tests whether the behavior of an institution, actor, or system is sensitive to a particular characteristic [30, 46] (e.g., whether the apparent gender of an applicant influences their position in a recruiter's results on a job search website [14]). The study design of algorithmic audits can also depart from that structure depending on the system tested and the behavior assessed [46], (see [44, 50]).

Algorithm audits have been performed in legal contexts. An investigation into risk assessment software used in criminal sentencing revealed racial disparities, with Black defendants nearly twice as likely to be falsely predicted at high-risk of re-offending [27]. This algorithm did not implement a law – its scores were based on a set of questions, which included sections about the defendant's leisure activities and attitudes [27].

Existing algorithm audits are primarily focused on finding bias, so do not operationalize correctness of the systems. Applying the legal guidelines, any given household can be pronounced definitively eligible or ineligible for benefits. The law treats households differently; it is not a bias of the screening tool when the pregnancy of a household member affects an eligibility prediction, it is a feature of the law.

Automated software testing systems could find potential issues with screening tools. Testing a piece of software can involve generating all possible combinations of input test cases to evaluate the software behavior [42]. However, such brute-force methods could overwhelm and bring down the online screening tools. Instead, methods that reduce the number of test cases by randomly selecting a subset of test cases while maintaining the diversity of generated test sets [15] or by searching for a subset of test cases that maximizes coverage of branches in the software [23] are more appropriate. However, they still do not test for realistic inputs.

Though lawyers employed by the agencies administering public benefits may be able to assess an applicant's eligibility flawlessly, this skill does not necessarily translate into expressing software requirements or creating a comprehensive test suite for software applications. Legal texts often require expertise to understand, so software engineers may have trouble encoding statutory language in computer code. Each profession has its own terminology and processes, so developer-lawyer collaboration is challenging and may not be effective without additional investments in cross-training [10]. It is difficult for lawyers and developers, either alone or together, to verify that these tools accurately implement benefits law before releasing them to the public.

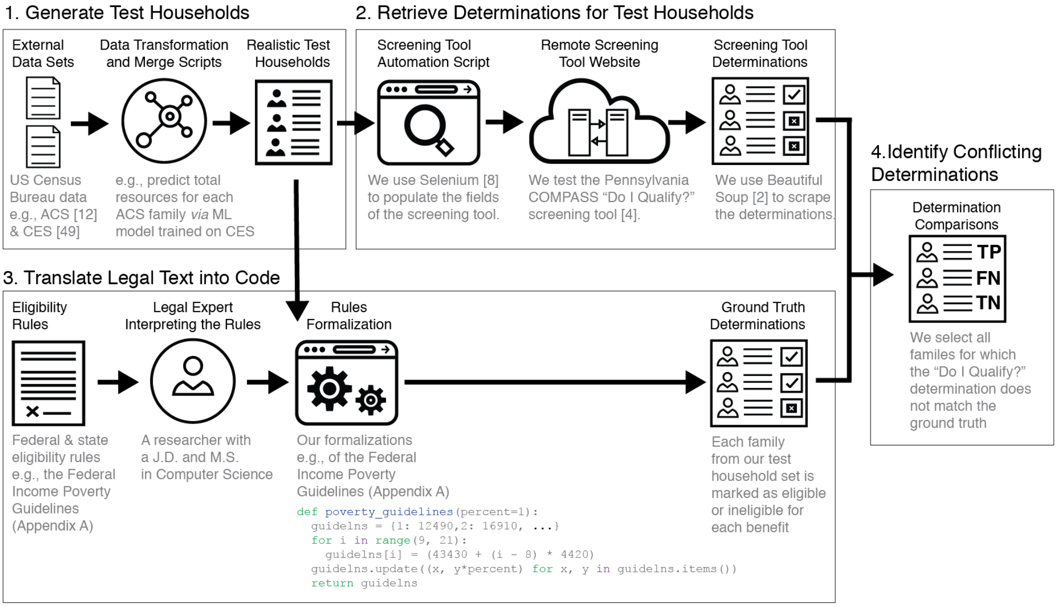
3 METHOD FOR AUDITING SCREENING TOOLS

Our method audits the correctness of online benefits screening tools on generated, but realistic, test inputs. We describe the general method that can be used to audit any public benefits screening tool in this section, and then detail our implementation for the Pennsylvania "Do I Qualify?" tool in section 4. To perform the audit, our method prescribes the following steps (Figure 1): 1) generate test households, 2) retrieve determinations for test households, 3) translate legal text into programming code, and 4) identify conflicting determinations.

3.1 Generating Test Households

Realistic testing data inputs allow us to focus our audit on the impact of the tool on real people. However, realistic test data require representative data. We identify the set of questions that appear on the screening tool. To generate our test set of inputs that match screening tool questions, we collect datasets that are representative of the potential applicants (e.g., if operating at the federal level, a national sample would be appropriate).

When a variable in the dataset is encoded in a format different from what the question expects, we transform the variable. For example, if the screening tool asks for the household's monthly income, but the data provides yearly income, we would divide yearly income by twelve. When variables span multiple datasets, we merge them by combining all variables that correspond to the screening tool questions into a single dataset. We manually match variables that overlap between the datasets and estimate the remaining variables of interest using machine learning.


1. Generate Test Households
External Data Sets
Data Transformation and Merge Scripts
Realistic Test Households
US Census Bureau data e.g., ACS [12] & CES [49]
e.g., predict total resources for each ACS family *via* ML model trained on CES
2. Retrieve Determinations for Test Households
Screening Tool Automation Script
Remote Screening Tool Website
Screening Tool Determinations
We use Selenium [8] to populate the fields of the screening tool.
We test the Pennsylvania COMPASS "Do I Qualify?" screening tool [4].
We use Beautiful Soup [2] to scrape the determinations.
4.Identify Conflicting Determinations
Determination Comparisons
TP
FN
TN
We select all familes for which the "Do I Qualify?" determination does not match the ground truth
3. Translate Legal Text into Code
Eligibility Rules
Legal Expert Interpreting the Rules
Rules Formalization
Ground Truth Determinations
Federal & state eligibility rules e.g., the Federal Income Poverty Guidelines (Appendix A)
A researcher with a J.D. and M.S. in Computer Science
Our formalizations e.g., of the Federal Income Poverty Guidelines (Appendix A)
Each family from our test household set is marked as eligible or ineligible for each benefit

```
def poverty_guidelines(percent=1):
  guidelns = {1: 12490,2: 16910, ...}
  for i in range(9, 21):
    guidelns[i] = (43430 + (i - 8) * 4420)
  guidelns.update((x, y*percent) for x, y in guidelns.items())
  return guidelns
```

Fig. 1. Flowchart of our method for auditing government benefits screening tools in four steps, with implementation examples for each step. 1) Generate test households from representative data sources that correspond to the screening tool questions, 2) input the test households to the screening tool and record the returned determinations, 3) input the same test households to a formalization of the eligibility rules produced by a legal expert to produce ground truth determinations, and 4) compare the determinations produced by the screening tool and the ground truth determinations to find conflicts.

3.2 Retrieving Determinations for Test Households

We submit each test household to the screening tool and record the determination. Manually filling out the online screening tool questions takes time and is error prone. Instead, our method uses web browser automation to simulate a tireless, faultless user's interaction with the screening tool. As we are auditing for errors in the screening tool, we do not wish to introduce human data-entry error that could confuse the source of problems.

To automate this step requires a script that can automatically open a web browser, navigate to the screening tool URL, enter household information to the tool, and scrape the returned results for each test household. This means that the script requires manual tuning to a screening tool's particular format to target the correct screening tool fields by matching input variables to question HTML elements, navigate the screening tool pages by clicking on links and buttons, and parse the determinations contained in HTML elements on the determination web page.

3.3 Translating Legal Text into Programming Code

Legal sources, such as statutes and policy manuals, establish the eligibility rules for each benefit. Assessing benefits eligibility for each test household individually would be tedious and could introduce human error. To automate labeling each household with ground truth, our method requires a formal representation of the eligibility guidelines to be translated into computer code.

Work from the field of computational law establishes that it is possible to encode legal rules in computer code [47]. Though some areas of law are harder to formalize, experts can hard-code laws for "single domains that are governed by a specific set of rules," [32] (e.g., as tax rules are encoded in tax-preparation software). Public benefits eligibility guidelines fall within this category. Although the rules only permit a single correct interpretation, this still requires a legal expert to help with rule formalization.

In our method, each formal representation of the eligibility guidelines must accept a household as input and, as output, return whether the household qualifies for the benefit. For instance, if a policy manual sets a $25,000 income threshold to qualify for a given benefit, the formal representation encodes the threshold and disqualifies a household if its income is in excess. We attempt to use the same fields solicited by the audited tools in our formalization to incorporate different legitimate motivations for screening tool design when assessing its accuracy. We input each of the test households into all formal benefits models and store predicted determinations. The formal models abstract away the messiness of passing user input and predictions between a browser and a potentially proprietary back-end.

3.4 Identifying Conflicting Determinations

Any difference between determinations from the screening tool and our formalization suggests a potential error with the screening tool. For example, when the formal model predicts eligibility, but the screening tool does not, households that deserve benefits receive advice not to apply and may not pursue the assistance they are entitled to receive. When the formal model predicts ineligibility, and the screening tool eligibility, households may fill out unnecessary paperwork.

However, simply totaling the number of errors does not capture the extent of harm. Instead, we compute the number of false negative determinations (i.e., the screening tool predicts a household ineligible for a benefit, but the formalization predicts it eligible) and the number of false positive determinations (i.e., the screening tool predicts a household eligible, but the formalization does not). This allows us to disentangle errors that are associated with different harms in our analysis.

Each wrong determination that predicts that a test household is ineligible for benefits represents a potential real household that did not get the help it deserved. Examining the composition of these households can help describe who is affected by screening tool errors. Thus, we also explore the data to find common reasons for incorrect determinations.

4 PENNSYLVANIA COMPASS “DO I QUALIFY?” SCREENING TOOL AUDIT

We illustrated our method on the Pennsylvania COMPASS “Do I Qualify?” screening tool [4]. COMPASS is the state’s online system for accessing many public benefits. At the center of the COMPASS homepage, there are two large buttons: “Apply Now” and “Do I Qualify?” The “Do I Qualify?” option takes the user to an online benefits screening tool. Unlike “Apply Now,” it does not demand contact information, and there are only three steps shown in the progress bar (compared to eleven different steps in “Apply Now”) [4]. Although “Apply Now” leads to the full, official benefits application, the “Do I Qualify?” option appears equally authoritative.

“Do I Qualify?” provides eligibility predictions for five different benefits: health care coverage, cash assistance, supplemental nutrition assistance, free or reduced price school meals, and subsidized child care. Heating assistance screening is offered seasonally; we performed data collection in September 2019, so we did not record data on this benefit.

After selecting the “Do I Qualify?” button, users choose the benefits they are interested in. The next page collects household-level information, including the name, age, and sex of each household member. The tool appends a separate information page for each member entered. After the user clicks through each household member’s page, the tool displays eligibility predictions for the household. The user can click a “More Information” link below each eligibility prediction to bring up a modal overlay that contains additional details about program requirements. An applicant need not certify that entered information is correct, which allows us to run the tool on numerous different inputs without rounds of agency approval.

4.0.1 Ethical Considerations. As the system we study intends to serve vulnerable populations, we were careful to conduct our research to minimize any additional frustrations or hazards. We submitted our study protocol to the University of Michigan Institutional Review Board (IRB) and, additionally, consulted the Office of the General Counsel. The IRB granted our study exempt status (HUM00158181) under research involving publicly available datasets. We assessed the risks of our method to three different stakeholder groups: the households whose information we use to generate our test dataset, the users who interact with the COMPASS website, and the Pennsylvania Department of Human Services which maintains the COMPASS screening tool.

To generate test households, we used datasets that were representative of the Pennsylvania population without exposing personal information from individual families. As we describe in detail below, our test families are based primarily on US Census data, which has strong confidentiality and privacy protections. We used Machine Learning to append additional fields to each simulated family, which potentially further distances the test households from the original survey respondents. Though we believe that our research context justifies our machine learning application, we do not plan to release our models, which predict sensitive health information.

Our method requires web scraping to record determinations from the screening tool. As our request for the website source code was denied, we interacted with the hosted site [4]. We could not locate any terms of use for the website or a `robots.txt` file, which site owners can post to instruct web crawlers about permissible interactions. We took steps to prevent overwhelming the servers with repeated requests and causing interruption for Pennsylvanians accessing the state's online services. Our scraper maintained at most ten connections at a time with the website, simulating ten simultaneous users. We paused between submitting each household, and froze all connections when we detected a slowdown in the COMPASS website response rate. Given that the screening tool is meant to serve the state of Pennsylvania, with an estimated population of 12.8 million people [3], we did not judge there to be a threat of service interruption from the submission of our test set.

Uncovering errors could undermine trust in e-government services, but our method provides a path to correct the errors we identified. The Pennsylvania Department of Human Services is responsible for providing crucial benefits to the state's residents, and any threat to the delivery of those benefits to eligible people should be addressed. We believe that transparency about errors in the screening tool is necessary for the department to fulfill its responsibilities.

4.1 Generating Test Households

A brute-force method is impractical here; there are too many possible cases to check them all. The tool permits households with up to twenty members, and each member page includes several binary and discrete fields. Also, we interacted with a hosted website, so the network constrained how quickly each response could be returned. Because it is more important that real applicants receive the correct results than that all edge cases are tested, we did not prioritize extreme cases (e.g., setting a household member's age to 150 or submitting households with college-educated infants). This also reduced unnecessary load on a public screening tool that serves real people.

We sourced test households from the subset of Pennsylvania state data in the American Community Survey (ACS) Public Use Microdata Sample [12]. This survey is collected annually by the United States Census Bureau, and we used the most recent version, from 2017. The dataset contains records at both the household and person levels, spanning thousands of variables from utility costs to veteran status. Each row in the household file corresponds to one household, and each row in the person file describes an individual bound to one of those households. We also used this source because it contains most of the data requested by “Do I Qualify?”; there is not a public, representative dataset that includes all fields.

There were four questions in the screening tool that could not be answered based solely on the information in the ACS, so we supplemented the data with additional datasets. The values we submitted for the household’s total resources, whether a member has undergone an operation that prevents pregnancy, whether the household had out-of-pocket medical expenses, and whether a member has aged out of foster care are estimates. For series, we select the surveys conducted in the year closest to 2017, when the ACS version used was collected. We used the Consumer Expenditure Survey from 2013 [49] for total resources of the household, the National Health Interview Survey from 2013 [37] for whether a member had undergone a hysterectomy, and the Current Population Survey: Annual Social and Economic Supplement Survey from 2017 [41] for whether the household had out-of-pocket medical expenses.

To combine the datasets, we manually matched the columns present in both the ACS and the supplemental survey. We then trained a machine learning model where the input variables are the matched columns from the supplemental survey and the output variable is the missing field. We used the model to predict the fields missing from the ACS for each household. For continuous data (e.g., total resources) we used linear regression, and for categorical data (e.g., if a household member had a hysterectomy) we used Random Forest classifiers.

To find the best model, we used nested cross validation on the training data from the surveys we used to supplement the ACS. We picked regressors with parameters that had the lowest Root Mean Squared Error (RMSE) and classifiers with parameters that had the highest F-1 score on our target test households (e.g., for total resources, we target households that make less than $25,750, the poverty line for a family of four). We then trained the best regressors and classifiers on the whole data from the supplemental datasets.

We trained a model based on National Survey of Child and Adolescent Well-Being [19] to predict whether a household member had formerly aged out of foster care but did not use it. The F-1 score was very low, likely due to the small sample size of aged-out foster youth in this survey. Instead of using the machine learning model, we combined aggregate statistics from the Kids Count Data Center [7], which provides the number of children who age out of foster care every year for each state, and American FactFinder [3], which provides state population estimates for different age groups. We divided the number of children aging out of foster care in Pennsylvania from 2011-17 by the total population aged 18 to 24 in Pennsylvania; about 0.45% of people in that range aged out of foster care. In our Python script, we set former foster care status with a 0.45% probability for people in the corresponding age range. Since former foster care youth are more likely to experience worse outcomes, such as unemployment and homelessness [7], and would thus qualify based on other household information, the random assignment approach tests the effect of the variable on a wider spread of individuals and could provide more information about the accuracy of the eligibility implementation with respect to this variable.

We then ran a Python script that transforms the ACS data, supplemented by our models to fill in missing variables in ACS, to match screening tool field formatting. For example, the school-level variable in the ACS is an integer with each value indicating a grade. The screening tool includes a drop-down field for school-level where grades are bucketed into options like “High School” and “Elementary School.” This mismatch required recoding the ACS variables, with several grade values pooled into one of the screening tool options. When our script finished, our household data aligned with the screening tool fields.

4.2 Retrieving Determinations for Test Households

Potential applicants using the “Do I Qualify?” screening tool must list household members, enter household information, and describe attributes of individual household members. We automated the entry of this information.

With Selenium [8], a browser automation tool, our data entry script used transformed household records to populate the fields of the screening tool. We manually matched our test variables to their corresponding “Do I Qualify?” HTML form elements. After the questions on a page were filled in, our script clicked the “Next” button.

After we submitted the household information, the screening tool presented a page with a list of the household’s eligibility determinations for selected benefits. We scraped the results page using Beautiful Soup, a Python library, [2] and stored the determinations (eligible or ineligible) for each of the benefits. We parallelized these operations. Our scripts ran ten processes simultaneously. On average, it took 11.5 seconds to evaluate one test household.

4.3 Translating Legal Text into Programming Code

To generate our ground truth, we applied the legal eligibility guidelines to each household. A researcher with a J.D. and M.S. in Computer Science produced a formalization of the eligibility guidelines for each tested benefit. Eligibility requirements are enumerated across several sources, as the “Do I Qualify?” screening tool covers a range of aid programs which are established by different levels of government. Across the five benefits, the legal sources contain hundreds of pages. Though not all of the information is relevant to the benefits eligibility rules, there is no compiled list of the applicable provisions, so we examined every section for relevance. We compiled the eligibility guidelines for each benefit from the statute or policy manual and encoded them into a Python function that accepts a household as input and returns an eligibility prediction as output.

For instance, we built a formal model of the Supplemental Nutrition Assistance Program (SNAP) benefit [40]. The food assistance requirements are scattered throughout the thirty-two chapters of the SNAP manual [40]. Chapter 540 contains the income and resource limits for SNAP eligibility. Five different tests can be applied to a household; selection depends on income level, the presence of an elderly or disabled member, and whether a member has been previously disqualified or sanctioned for violating program requirements.

We formalized the tests for SNAP in Python code. For example, one test for “households having an elderly/disabled member” set an income limit of “200% FPIG” (Federal Poverty Income Guidelines); the corresponding conditional is `household['HINCP'] < 200 * fpig(household['NP'])`, where `household` is an object that contains household information, the ACS variables `HINCP` refers to household income and `NP` refers to number of persons, and `fpig()` is a function that accepts the number of people in a household and returns the poverty guideline cutoff (see Appendix A for the full function definition and legal source).

Unlike the “Do I Qualify?” screening tool, our formal models isolate eligibility rules. They strip away the complexity of presenting a user interface and passing values between the front-end and back-end. As we did not receive the code for the API endpoint that responds with screening tool determinations, we do not know how the “Do I Qualify?” tool is actually implemented. However, we did not need this information to apply our method.

The screening tool would be much longer if it required all the information needed for evaluation under the full guidelines. From the fields the tool collects, we constructed the most inclusive formalizations possible. However, we noted when the tool did not solicit a field that would extend categorical eligibility, rather than assuming that all potential applicants fall into that category.

Migrant children, for example, are eligible for free school lunches (per 7 CFR §245.6(b)(8)); the tool does not solicit this information, but we do not assume all students are migrant children.

4.3.1 The Role of Solicited Fields Absent from Formal Models. The "Do I Qualify?" tool requests some information that overreaches the requirements of Pennsylvania benefits eligibility handbooks. For instance, when users indicate interest in cash assistance programs, a question is conditionally posed to female household members older than nine and younger than sixty-one: "Has {Individual.FirstName} undergone an operation that prevents pregnancy?" We could not find a corresponding requirement in the cash assistance policy manual for Pennsylvania. One program, SelectPlan [39], offers family planning services to women aged eighteen to forty-four, but its requirements are set out in the medical assistance, not cash assistance, handbooks.

We added additional test data to probe this question. Data on operations that prevent pregnancy was absent from the ACS and was supplied by our machine learning models. We tested the effect of this question in the screening tool by comparing households where we predicted that a member had undergone an operation that prevents pregnancy to otherwise identical households where the same member did not. With our augmented dataset, we could isolate the impact of this question on receiving benefits. As this technique increases the size of the test dataset, we used it sparingly. We performed it only for fields in the screening tool that do not have an analogue in the formal representation of the eligibility guidelines.

4.4 Results Analysis

We submitted a test set of 68,983 households to the "Do I Qualify?" screening tool. We selected to screen for all benefits when processing each household. Each benefit has its own set of rules, which may be offered to different subsets of members from the same household. We examine each benefit separately.

4.4.1 Child Care Works. The screening tool predicted that every household we submitted was ineligible for the subsidized child care benefit. The Child Care Works benefits guidelines are established in the Pennsylvania Code Title 55, Chapter 3041, titled "Subsidized Child Care Eligibility" [16]. It permits applications from families, defined as a parent or caretaker, their spouse, and their children. Families with a child who needs supervision and whose income falls below 200% of the Federal Poverty Income Guidelines (FPIG) are eligible.

Our formalization first split households into family units. It then checked each family unit for the presence of an eligible child; children are eligible until age thirteen, and that cutoff is extended until age nineteen for children with disabilities. The formalization also verified that there were no adults available to supervise the child; all parents must either be employed or have a disability. We calculated appropriate deductions, such as selecting the Stepparent Deduction from 55 Pa. Code §3041 App. C if either the caretaker or spouse is a step-parent of the eligible child. We totaled the family income, subtracted deductions, and compared that value to the FPIG threshold for the family size.

Our most restrictive interpretation of the guidelines predicted that 3,198 households (4.6%) are eligible for this benefit. This formalization required that all parents had positive `job` or `disability` values. To validate there was no error in our scrape, we verified that these families were denied benefits by the screening tool by manually submitting ten test households which our formal model predicted eligible. All ten were predicted ineligible for Child Care Works.

We created a secondary formalization that is also suitable for a screening tool implementation. Per §3041.44, a family in which a parent who does not currently have a job but has secured prospective work may still be eligible for this benefit. Normally, if there is an unemployed adult present, the expectation is that they will look over the children. However, since childcare costs may be the factor most contributing to the adult's decision to stay home, this second formalization also recommends the benefit to otherwise eligible families that have an unemployed adult without disabilities. This formalization predicts 4,641 households (6.7%) eligible. Again, the screening tool predicted that zero (0%) of households were eligible for Child Care Works.

4.4.2 Free and Reduced Price School Lunch. Child nutrition programs offer free or reduced price lunches to students from low-income families. The income eligibility guidelines are set at the federal level; the Federal Register Vol. 84, No. 54, declares the thresholds for different household sizes for the period from July 1, 2019, through July 30, 2020. The general requirements, applicable across annual periods, appear in 7 CFR §245.6. These extend categorical eligibility to foster children, homeless children, migrant children, runaway children, and children in Head Start programs.

Our formalization predicted 3,310 households (4.8%) are eligible to receive this benefit. We first verified that there was a child present in the household. We then totaled household income and disqualified any household in excess of the income eligibility threshold for its size. This approach assumed that the potential applicant entered income information in accordance with the federal definition, i.e., excluding the value of free lunches received and any other sources of federal income that are statutorily exempt.

The screening tool predicted 4,230 households (6.1%) eligible for this benefit. Of the 939 households the screening tool predicted eligible that our formalization did not, 334 had six-figure household incomes. In one extreme case, the screening tool predicted that a family of five (a married couple and their three biological children) who had an annual household income of $828,500 were eligible to receive free or reduced price school meals; the Federal Register sets the income threshold for a household with five members at $55,815. This result was verified and video-captured by a researcher who manually entered the household's information into the screening tool.

Using relationship values from the ACS, our model also identified school-age foster children and marked their households as eligible. This information was not solicited by the screening tool, though foster children are categorically eligible under §245.6(b)(8). As the variable was present in our dataset, we included it to test the effect of its omission.

99.4% of families that our formalization predicted eligible for free or reduced price lunch were also predicted eligible by the screening tool. There were sixteen families who did not qualify for this benefit according to the income eligibility guidelines, but include foster children. As the screening tool does not solicit information about foster children, or other categorically eligible groups, it predicted that all of these households were ineligible. The "More Information" modal window for this benefit on the "Do I Qualify?" eligibility predictions page does not include information about categorical eligibility for foster children.

4.4.3 Supplemental Nutrition Assistance Program. The Supplemental Nutrition Assistance Program (SNAP) provides low-income households with payment cards that can be used to purchase approved food items. It is a federal aid program, authorized by the Food and Nutrition Act of 2008 [1], but benefits are administered by states. The Pennsylvania Department of Human Services Office of Income Maintenance publishes a SNAP manual that includes eligibility guidelines. These rules define a household as "people who live together and buy and prepare meals together" [40]. Under this definition, there can be multiple households living in the same housing unit.

The "Do I Qualify?" screening tool does not prompt potential applicants to indicate which household members prepare food together. Our model incorporated the tool's assumption that households submitted prepared meals together. We discuss this assumption in section 5.3.2.

The SNAP manual provides five different tests to establish household eligibility in §540.1. The first test is applied to households with an elderly or disabled member; the income limit is 200% of the FPIG. If no member is elderly or disabled, the second test sets an income limit of 160%. Our formalization checked for the presence of an elderly or disabled member, and compared household income against the proper threshold. We predicted any household satisfying these tests eligible for SNAP. If households did not pass either test, a third test is applied, which permits several deductions from income to determine a new, net income. To be eligible under this third test, a household's net income must not exceed 100% of the FPIG and their countable assets (e.g., cash, vehicles) must not be worth more than $3,500.

There are many allowable deductions (e.g., medical expenses, child support payments) in calculating net income which are not solicited by the screening tool. So, in addition to the formalization which applies the first two tests, and is thus underinclusive, we used a formalization that applies the asset limit from the third test, and is thus overinclusive. Our model also did not incorporate the two more restrictive tests that are used when a household member has previously violated program requirements; they set lower income and asset limits—anyone who qualifies under the last two tests will also qualify under the first two.

The screening tool predicted 21,071 households (30.5%) eligible for SNAP. Our underinclusive formalization predicted 21,568 households (31.3%) eligible. Our overinclusive formalization predicted 26,945 households (39.1%) eligible. All households predicted eligible by the screening tool are also predicted eligible by both of our formalizations. If a potential applicant screens for the SNAP benefit alone, the tool does not request information about household assets. Based on this, as well as our results, the screening tool does not appear to apply the asset limit from the third test even when household asset information was solicited.

Trends emerge in the households that we predicted eligible, but the screening tool did not. Per §550.51, household income should exclude earned income from a child who is seventeen or younger, attends school, and lives with a parent. One of our test households has four members; the two parents have a combined annual income of $39,000, their sixteen-year-old son has an annual income of $5,000, and their eleven-year-old son does not have a job. For a family of four, 160% of the FPIG is $41,200. The screening tool predicted that this family was ineligible for SNAP.

However, when a researcher manually entered the same family, but reduced the sixteen-year-old's income to $0, the tool predicted the household was eligible. This indicates that the "Do I Qualify?" screening tool incorrectly takes into account child income, which can inflate household income such that the tool predicts eligible households are ineligible.

4.4.4 Cash Assistance. The Pennsylvania Department of Human Services Office of Income Maintenance offers a Cash Assistance policy manual [38]. Though it forms the basis of our formalization, it is critically out of date. In Chapter 106, the policy manual lays out the eligibility requirements to receive General Assistance. Under this program, extremely low-income people who are temporarily or permanently disabled receive monthly payments of around $200. This chapter states the resource requirements (less than $250 for individuals, $1000 for families), and forms of proof for disability. It does not include the updated language from 2019 Act 12: "the general assistance cash assistance program shall cease August 1, 2019" [9].

Thus, our formalization covered only the Temporary Assistance for Needy Families (TANF) provisions of the Cash Assistance policy manual.

The "Do I Qualify?" screening tool predicted 2,266 households (3.3%) eligible for cash assistance. However, that tool seems not to implement the full provisions permitting child-only budget groups, which is contemplated by Chapter 110. Grandparents raising grandchildren are eligible to receive assistance even when their income exceeds the thresholds, as they are not mandatory budget members, per §110.4. "Do I Qualify?" predicted these grandparent-headed households from our test set ineligible.

The full text of the "More Information" modal provides little additional information:

> Common reasons why people do not qualify for Cash Assistance
> Does not meet one of the following requirements: pregnancy or needy family [4]

It does not link to the Cash Assistance handbook or indicate that there are nuances to the definition of "needy family" which expand eligibility beyond the prediction logic of the screening tool.

4.4.5 Medical Assistance. Per the manual, there are twenty-two different categories that describe people eligible for medical assistance [39]. The screening tool predicted almost every household eligible for health care coverage. Of the 68,983 test households submitted to the screening tool, only 380 households (0.55 %) were predicted ineligible.

There were still errors. Households with members over sixty-five were deemed ineligible for COMPASS coverage and directed to the Health Insurance Marketplace. This recommendation is inappropriate, as the federal healthcare program Medicare provides healthcare for people over sixty-five and is not part of the Health Insurance Marketplace [6].

4.4.6 Impact of Fields Absent from Eligibility Guidelines. We found no impact of the question "Has {Individual.FirstName} undergone an operation that prevents pregnancy?" on benefits eligibility. Of the 9074 households that contained a member who had undergone a hysterectomy, 159 were predicted to receive cash assistance, 1346 for SNAP, 9013 for health assistance, 228 for free or reduced price school lunch, and 0 for subsidized child care. When we resubmitted these households with no members having undergone an operation to prevent pregnancy, the results were the same.

5 DISCUSSION

We first situate our auditing method in the larger context of the American welfare ecosystem. We then turn to the results of our audit of the Pennsylvania COMPASS "Do I Qualify?" screening tool, investigating both how these errors are patterned across determinations and how code artifacts in these tools disparately impact different groups. These results motivate policy recommendations towards improving the reliability of future screening tools for public benefits in the United States.

5.1 Defective Screening Tools and Benefits Administration

Screening tools are a component of a broader welfare system. When they produce erroneous predictions, there are consequences beyond the immediate interaction between user and interface. Different mistakes can have different impacts; under-inclusion can materially deprive eligible households, and over-inclusion can undermine the legitimacy of welfare programs. Agencies are reluctant to open their code for inspection, citing security as a concern. Our auditing method can be used to identify both classes of error without accessing government source code. Further, exploration of the results produced by our method can help identify affected classes of people, who can then apply political pressure to institutions as a group rather than as disconnected individuals.

Individuals who are advised that they are ineligible for benefits may be discouraged from filling out a full application. Our method found many examples of households with exceedingly limited financial resources who were given wrong determinations. No families were advised that they were eligible for subsidized child care. Thousands of families that our formalization found eligible for benefits instead received predictions of ineligibility from the "Do I Qualify?" tool.

False positives can also be harmful. If a potential applicant is consulting the tool in good faith, an erroneous prediction of eligibility can waste their time. For households in borderline situations, it can be disappointing to go through the stressful and invasive processes of applying for benefits, only to be denied aid. On the other hand, advising wealthy clients that they are eligible for public benefits can undermine the legitimacy of the welfare system. One millionaire who applied for and received SNAP benefits used his story to call for tighter restrictions on the program in front of a legislative body [18]. Our method reveals that many instances of wealthy, and clearly ineligible, households were predicted to be eligible for cash assistance and SNAP. These types of errors can contribute to the perception that public benefits are given to people who do not need them.

When our auditing method exposes an error, we know that a rule has been broken. Though uncovering an instance of error is important, one data point does not indicate the extent or source of the problem. By testing many examples, we can find patterns of error that indicate which rule has been broken. Knowing which provision has been incorrectly implemented can help affected individuals articulate specific demands for specific fixes, as well as identify a class of others who may also be injured. For example, grandparents responsible for their minor grandchildren have shared experiences and means of communicating with each other, such as through Facebook groups [22]. If made aware that the "Do I Qualify?" tool does not accurately predict cash assistance eligibility for grandparent-headed households, they could spread the news and act in concert to demand more accurate screening tools.

Government agencies that administer public benefits programs are protective of their computer code, but technology that controls access to vital services ought to be correct. An audit that exposed their source code could verify the results of trillions of test cases and inspect rule encodings line-by-line. Thus, our auditing method is not an ideal solution. We cannot promise that we expose every source of error, and scraping a website can tax resources. However, it can be run by private citizens and organizations without government authorization, and it does show some error.

When an audit indicates that a screening tool correctly implements the law, the method does not lose its usefulness. As eligibility rules frequently change, a screening tool can be assessed by re-running the project at intervals. This would be useful after the yearly release of the poverty guidelines by the Department of Health and Human Services [36], since state benefits eligibility often tracks the national rules. Depending on how the code is maintained, the screening tools may issue incorrect determinations in the future.

5.2 Disparate Impacts on Applicant Groups

Households do not receive equal treatment from the "Do I Qualify?" tool. We saw this not just in eligibility predictions, the tool behavior, and code artifacts. Though not the focus of our investigation, we observed aspects of the "Do I Qualify?" tool separate from its predictions that provided reason to question its neutrality. We noted issues concerning sex, gender, and sexual orientation.

The form solicits the sex of each household member, permitting (and requiring) the selection of "male" or "female." Pictograms are added beside the names on the person information tabs; females wear skirts, and males wear pants. If the "female" option is selected for a household member, the tool conditionally display questions relating to pregnancy. In the Angular front-end code, the variable that holds this value is `gender`.

The "Do I Qualify?" screening tool requires same-sex couples to perform more work than different-sex couples. When someone in a different-sex marriage selects the "Husband" or "Wife" option from the relationship drop-down, fields are automatically populated for the other spouse. For example, if a female household member Ada selects "Wife" as her relationship to male household member Ali, the personal information page for Ali will have the "Husband" selection auto-selected for his relationship to Ada. This behavior does not occur for same-sex couples, who do not have their relationship automatically recognized. The relationship option must be manually selected on both spouses' person information pages.

In the section of the form where information is solicited about individual household members, a question is conditionally posed to females older than nine and younger than sixty-one: "Has {Individual.FirstName} undergone an operation that prevents pregnancy?" This question appears if the applicant selects to screen for cash assistance. We could not find a cash assistance provision related to this question. Also, our method did not detect any information gain from answering this question affirmatively – the households we submitted received the same predictions whether a female member indicated "yes" or "no" to this question.

The phrasing of this question is insensitive to its context. Eugenics has been posed as a solution to poverty; even today, controversial organizations offer cash payments to women in exchange for voluntary sterilization [28]. With the complexity of the benefits eligibility guidelines, it is wasteful and stigmatizing to include it as one of eight questions posed on the person information pages.

5.3 Recommendations for Screening Tool Implementations

Our research shows that screening tools can make errors. We provide a set of recommendations for reducing error, as well as communicating the tool limitations to potential applicants.

5.3.1 Fidelity to Legal Requirements and Managing Ambiguity. Delegating the administration of law to software can increase the efficiency of evaluating benefits applications. It can also be wrong. Moving from natural language to formal language is not an automatic process.

Unlike laws that are open to interpretation or intentionally delegate administrative discretion, eligibility guidelines are bright-line rules. A household below an income threshold is eligible, a household above is not. Such rules can be represented exactly in computer code, but may require a legal expert to identify relevant provisions and to convey the meaning of text to a software engineer. To reason correctly about a household's eligibility, a system must encode the full set of rules.

However, screening tools do not always encode the full set of eligibility rules. Other considerations, such as convenience, encourage simplification. While this can make the tool less daunting for a potential applicant, simplification leads to ambiguity. When the screening tool does not solicit a variable that is needed to evaluate the status of a rule, its logic is muddied. If a household member cannot indicate on a screening tool that they are a foster parent, the resulting prediction must either disregard or extend to all households the specific provisions that refer to foster families. The designers of the tool can intentionally select which questions to include or omit based on the strategies available to resolve imprecision. Arguably, the tool logic should be structured so that everyone conceivably eligible under the limited set of questions is predicted eligible.

To decide the appropriate balance between correctness and convenience, software engineers and legal experts should work together. Software engineers write the tool logic that determines how ambiguity is resolved, but legal experts might have a better understanding of how the trade-offs will affect the applicant populations. If not explicitly negotiated, the simplification of the rules could be determined arbitrarily.

We echo recommendations for cross-training lawyers and software engineers [10]. Lawyers need to understand how code works to articulate effectively how the internal logic of the tool should operate. Software engineers need to learn about the law so they can ask lawyers for clarification and write comprehensive test suites. Together, they should compile a set of test input and expected outputs to identify obvious errors.

5.3.2 Assumptions about Potential Applicant Knowledge. Screening tools should not assume that potential applicants know the different household composition rules. When the "Do I Qualify?" tool prompts a potential applicant to enter household members, it does not specify who should be included. This can affect eligibility predictions. Different benefits define different household composition rules, and applicant units may be smaller than the people living together in a housing unit [38, 40].

For example, for SNAP, the relevant unit is members of household who prepare meals together. There may be multiple applicant groups in one household, and each may submit a separate application for nutrition assistance benefits [40]. The "Do I Qualify?" tool neither defines "household" in text nor cues potential applicants to apply as a sub-unit when applicable. It evaluates every member of the household together. When one household sub-unit meets the eligibility requirements, but the full household does not, eligible people could lose benefits.

Screening tool developers should avoid making such assumptions about potential applicants' background knowledge. Questions should be structured so that they draw out components of compound concepts from the eligibility guidelines. Tools should guide and inform potential applicants so they do not miss out on benefits.

5.3.3 Communicating the Limitations of a Tool. An applicant may not know that screening tools provide an abridged version of the rules. While screening tools might include some language that indicates that the advice provided is not an official determination, they seem authoritative. For example, the Pennsylvania Compass "Do I Qualify?" screening tool carries over the same interface elements, such as icons and color scheme, as the full application, so there is no visual distinction between the two portals. Applicants rarely have expert knowledge about benefits eligibility, so they could defer to the judgement of a screening tool despite posted disclaimers.

The "Do I Qualify?" tool offers some explanations for the basis of its predictions, but they may not be enough to dispel the negative consequences of error. For example, below each prediction on the screening results page is a "More Information" button. Clicking on it brings up a modal window overlay packed with small text. It indicates that there are additional provisions that might affect eligibility, but does not direct people to their sources. The organization of the modal information is inconsistent across benefits. The Cash Assistance modal contains just 22 words and gives two reasons people do not qualify, while the Child Care modal contains 368 words and provides nine additional ways to qualify, five common reasons for not qualifying, and a hot-line number for more assistance. It is unclear that this presentation of information would prompt a potential applicant to dig deeper for a more accurate way to assess their eligibility.

The idea of a screening tool is appealing. The legal sources are long. They include provisions directed at institutions and administrators, which do not help applicants understand their rights or requirements. However, if the tool does not provide reliable predictions that helpfully inform potential applicants, it should not be pitched as an advisor.

5.3.4 Assessments of Screening Tool Usefulness. Given the practical limitations of developing and maintaining screening tools, agencies must determine whether imperfect screening tools are useful to potential applicants. For example, such screening tools only reach applicants with Internet access, but online administration may not benefit all groups equally (e.g., people who apply using paper applications) [34].

More study is needed to learn the extent to which screening tools filter out applicants. If potential applicants turn out to be undaunted by predictions of ineligibility and plunge forth into applications, errors might not be a serious drawback. However, if issuing incorrect predictions of ineligibility make potential applicants less likely to complete full applications, it may be better that they never consult the tool.

With restricted pools of money allotted to administering and establishing public benefits, funding screening tools and audits to ensure their accuracy might not be the best investment choice. More work is needed to determine whether screening tools are helping or harming applicants.

5.4 The Future of Online Benefits Administration

We find that there is a vast difference between what the law says, and what the tool does. The law on the books also warrants closer attention. As social scientists have pointed out, "[t]echnologies of poverty management are not neutral. They are shaped by our nation's fear of economic insecurity and hatred of the poor; they in turn shape the politics and experience of poverty" [21]. Existing technology and rules normalize the rigorous requirements and hidden pitfalls that shape the benefits applications process.

Means-testing is not the only policy available for distributing resources. Transition to universal programs that provide benefits to all residents would obviate the need for an audit like we propose. Verifying that screening tools are working correctly adds additional work to the grueling and expensive process of means-testing benefits applicants. If the United States were to remove barriers to access, we would not need to scrutinize the shape of such barriers.

6 CONCLUSION

As long as the US welfare system remains so complex, the tools meant to help people navigate it should be well calibrated. We do not know what potential applicants expect from an interaction with a screening tool, nor how they make sense of eligibility predictions.

While our auditing method could help produce more accurate screening tools, it alone will not address the structural reasons that these errors exist and have been tolerated. Screening tools are not the single broken piece in an otherwise perfect welfare system.

Errors in full benefits applications have caused real, demonstrable deprivation [13, 25]. Future work could explore auditing methods for automated application evaluation technology. This would require much greater institutional cooperation than we secured. Our method cannot be directly fitted to this technology, since filing false applications is fraud. Further, full applications are far more complex than screening tools and their records persist across interaction sessions.

Researchers in other disciplines have engaged with these issues and are exploring alternate ways to structure the relations between the state and the poor. They might not know what technology is available to advance those visions. In the future, we should not just be looking to verify poverty management tools and perfect existing institutional practices. Instead, we must interrogate the structural problems that lead to these broken systems. If we believe that technology can play a positive role in poverty management, we can imagine better systems for providing care.

ACKNOWLEDGMENTS

We would like to acknowledge the efforts of Ponette Rubio and Alvin Hermans, who helped with the construction of our machine learning models. Catherine Morse at the UM library assisted us with identifying relevant datasets. We are indebted to Dina Karageorgos, Jeffrey Lockhart, the members of CompHCI lab, and our reviewers for providing glittering feedback that helped to smooth our rambling into a presentable form. Thanks also to Jeffrey Bilik, for typing.

REFERENCES

[1] 2008. Food and Nutrition Act of 2008. Retrieved September 20, 2019 from https://legcounsel.house.gov/Comps/Food%20And%20Nutrition%20Act%20Of%202008.pdf

[2] 2015. Beautiful Soup Documentation. Retrieved September 20, 2019 from https://www.crummy.com/software/BeautifulSoup/bs4/doc/

[3] 2019. American FactFinder. Retrieved September 20, 2019 from https://factfinder.census.gov/faces/tableservices/jsf/pages/productview.xhtml

[4] 2019. COMPASS HHS Do I Qualify? Retrieved September 20, 2019 from https://www.compass.state.pa.us/Compass.Web/Screening/DoIQualify#/SelectBenefits

[5] 2019. DTA Connect - Massachusetts Department of Transitional Assistance. Retrieved September 20, 2019 from https://dtaconnect.eohhs.mass.gov/screening

[6] 2019. Health Insurance Marketplace. Retrieved September 20, 2019 from https://www.healthcare.gov/

[7] 2019. Kids Count Data Center. Retrieved September 20, 2019 from https://datacenter.kidscount.org/data/tables/5101-foster-care--youth-ages-18-20-aging-out-to-a-non-family-setting?loc=40&loct=2#detailed/2/any/false/1754,1718,1606,1538,1473,1472,1467,1471,1466,1470/any/11538

[8] 2019. Selenium - Web Browser Automation. Retrieved September 20, 2019 from https://www.seleniumhq.org

[9] Pennsylvania General Assembly. 2019. Act of Jun. 28, 2019, P.L. 43, No. 12. Retrieved September 20, 2019 from 'https://www.legis.state.pa.us/cfdocs/legis/li/uconsCheck.cfm?yr=2019&sessInd=0&act=12'

[10] Anna Bobkowska and Magdalena Kowalska. 2010. On efficient collaboration between lawyers and software engineers when transforming legal regulations to law-related requirements. *2010 2nd International Conference on Information Technology, (2010 ICIT)* (2010), 105–109.

[11] Anna Brown, Alexandra Chouldechova, Emily Putnam-Hornstein, Andrew Tobin, and Rhema Vaithianathan. 2019. Toward Algorithmic Accountability in Public Services: A Qualitative Study of Affected Community Perspectives on Algorithmic Decision-making in Child Welfare Services. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, Article 41, 12 pages. https://doi.org/10.1145/3290605.3300271

[12] United States Census Bureau. 2017. American Community Survey. Retrieved September 20, 2019 from https://www.census.gov/programs-surveys/acs/data/pums.html

[13] Robert N. Charette. 2018. Michigan's MiDAS Unemployment System: Algorithm Alchemy Created Lead, Not Gold. *IEEE Spectrum* (Jan 2018). https://spectrum.ieee.org/riskfactor/computing/software/michigans-midas-unemployment-system-algorithm-alchemy-that-created-lead-not-gold

[14] Le Chen, Ruijun Ma, Anikó Hannák, and Christo Wilson. 2018. Investigating the Impact of Gender on Rank in Resume Search Engines. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, Article Paper 651, 14 pages. https://doi.org/10.1145/3173574.3174225

[15] T. Y. Chen, T. H. Tse, and Y. T. Yu. 2001. Proportional Sampling Strategy: A Compendium and Some Insights. *J. Syst. Softw.* 58, 1 (Aug. 2001), 65–81. https://doi.org/10.1016/S0164-1212(01)00028-0

[16] The Pennsylvania Code. 2019. Subsidized Child Care Eligibility. Retrieved September 20, 2019 from https://www.pacode.com/secure/data/055/chapter3041/chap3041toc.html

[17] Indiana Supreme Court. 2016. State v. International Business Machines Corp.

[18] Don Davis. 2018. Minnesota millionaire tells lawmakers he got food stamps to make a point. *Pioneer Press* (April 11 2018). https://www.twincities.com/2018/04/11/minnesota-millionaire-tells-lawmakers-he-got-food-stamps-to-make-a-point/

[19] Melissa Dolan, Keith Smith, Cecilia Casanueva, Heather Ringeisen, HHS P2320062930YC, M Dolan, K Smith, C Casanueva, and H Ringeisen. 2011. NSCAW II baseline report: Introduction to NSCAW II final report. *Washington, DC: Office of Planning, Research and Evaluation, Administration for Children and Families, US Department of Health and Human Services* (2011).

[20] G. Esping-Andersen. 1996. *Welfare States in Transition: National Adaptations in Global Economies*. SAGE Publications. https://books.google.com/books?id=7UJsCgAAQBAJ

[21] Virginia Eubanks. 2018. *Automating Inequality: How High-Tech Tools Profile, Police, and Punish the Poor*. St. Martin's Press, Inc., New York, NY, USA.

[22] Facebook. [n.d.]. Grandparents Raising Grandchildren. Retrieved September 20, 2019 from https://www.facebook.com/Grandparents-Raising-Grandchildren-159724844105830/

[23] Mark Harman, S. Afshin Mansouri, and Yuanyuan Zhang. 2012. Search-based Software Engineering: Trends, Techniques and Applications. *ACM Comput. Surv.* 45, 1, Article 11 (Dec. 2012), 61 pages. https://doi.org/10.1145/2379776.2379787

[24] Paul T. Jaeger and John Carlo Bertot. 2010. Designing, Implementing, and Evaluating User-Centered and Citizen-Centered E-Government. *Int. J. Electron. Gov. Res.* 6, 2 (April 2010), 1–17. https://doi.org/10.4018/jegr.2010040101

[25] Francesca Jaroz, Heather Gillers, Tim Evans, and Bill Ruthhart. 2008. Rollout of welfare changes halted: Contractor takes too long to process food stamp requests, feds say; action is a blow to Daniels' privatization push.

[26] Jeff Johnson and Jonathan Lazar. 2010. E-Government: Services for Everyone, Everywhere, Eventually. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems (CHI EA '10)*. Association for Computing Machinery, New York, NY, USA, 3139–3142. https://doi.org/10.1145/1753846.1753936

[27] Surya Mattu Julia Angwin, Jeff Larson and Lauren Kirchner. 2016. Machine Bias. Retrieved January 15, 2020 from https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

[28] Rebecca M. Kluchin. 2009. *Fit to Be Tied: Sterilization and Reproductive Rights in America, 1950-1980*. Rutgers University Press. http://www.jstor.org/stable/j.ctt5hj13v

[29] Christopher A. Le Dantec and W. Keith Edwards. 2008. Designs on Dignity: Perceptions of Technology among the Homeless. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. Association for Computing Machinery, New York, NY, USA, 627–636. https://doi.org/10.1145/1357054.1357155

[30] Linfeng Li, Tawanna R. Dillahunt, and Tanya Rosenblat. 2019. Does Driving as a Form of "Gig Work" Mitigate Low-Skilled Job Seekers' Negative Long-Term Unemployment Effects? *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article Article 156 (Nov. 2019), 16 pages. https://doi.org/10.1145/3359258

[31] Lorna Lines, Oluchi Ikechi, and Kate S. Hone. 2007. Accessing e-Government Services: Design Requirements for the Older User. In *Universal Access in Human-Computer Interaction. Applications and Services*, Constantine Stephanidis (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 932–940.

[32] Nathaniel Love and Michael Genesereth. 2005. Computational Law. In *Proceedings of the 10th International Conference on Artificial Intelligence and Law (ICAIL '05)*. ACM, New York, NY, USA, 205–209. https://doi.org/10.1145/1165485.1165517

[33] Frederick B. Mills. 1996. The Ideology of Welfare Reform: Deconstructing Stigma. *Social Work* 41, 4 (07 1996), 391–395. https://doi.org/10.1093/sw/41.4.391 arXiv:http://oup.prod.sis.lan/sw/article-pdf/41/4/391/5311029/41-4-391.pdf

[34] Pippa Norris. 2001. *Digital Divide: Civic Engagement, Information Poverty, and the Internet Worldwide.* Cambridge University Press, New York, NY, USA.

[35] United States Department of Agriculture. 2019. Estimates of State Supplemental Nutrition Assistance Program Participation Rates in 2016. Retrieved September 20, 2019 from https://www.fns.usda.gov/snap/reaching-those-need-estimates-state-supplemental-nutrition-assistance-program-participation-rates-fy

[36] Department of Health and Human Services. 2019. Annual Update of the HHS Poverty Guidelines. Retrieved September 20, 2019 from https://www.govinfo.gov/content/pkg/FR-2019-02-01/pdf/2019-00621.pdf

[37] United States Department of Health, Human Services. Centers for Disease Control, and Prevention. National Center for Health Statistics. 2015. National Health Interview Survey, 2013. https://doi.org/10.3886/ICPSR36147.v1

[38] Pennsylvania Department of Human Services. 2019. Cash Assistance Handbook. Retrieved September 20, 2019 from http://services.dpw.state.pa.us/oimpolicymanuals/cash/index.htm

[39] Pennsylvania Department of Human Services. 2019. Medical Assistance Eligibility Handbook. Retrieved September 20, 2019 from http://services.dpw.state.pa.us/oimpolicymanuals/ma/index.htm

[40] Pennsylvania Department of Human Services. 2019. Supplemental Nutrition Assistance Program (SNAP) Handbook. Retrieved September 20, 2019 from http://services.dpw.state.pa.us/oimpolicymanuals/snap/index.htm

[41] United States. Bureau of the Census and United States. Bureau of Labor Statistics. 2018. Current Population Survey: Annual Social and Economic (ASEC) Supplement Survey, United States, 2017. https://doi.org/10.3886/ICPSR37075.v1

[42] Alessandro Orso and Gregg Rothermel. 2014. Software Testing: A Research Travelogue (2000–2014). In *Proceedings of the on Future of Software Engineering (FOSE 2014)*. ACM, New York, NY, USA, 117–132. https://doi.org/10.1145/2593882.2593885

[43] Kate Rabinowitz and Kevin Uhrmacher. 2019. What Trump proposed in his 2020 budget. Retrieved September 20, 2019 from https://www.washingtonpost.com/graphics/2019/politics/trump-budget-2020/

[44] Ronald E. Robertson, Shan Jiang, David Lazer, and Christo Wilson. 2019. Auditing Autocomplete: Suggestion Networks and Recursive Algorithm Interrogation. In *Proceedings of the 10th ACM Conference on Web Science (WebSci '19)*. Association for Computing Machinery, New York, NY, USA, 235–244. https://doi.org/10.1145/3292522.3326047

[45] Paul Robinette, Wenchen Li, Robert Allen, Ayanna M. Howard, and Alan R. Wagner. 2016. Overtrust of Robots in Emergency Evacuation Scenarios. In *The Eleventh ACM/IEEE International Conference on Human Robot Interaction (HRI '16)*. IEEE Press, Piscataway, NJ, USA, 101–108. http://dl.acm.org/citation.cfm?id=2906831.2906851

[46] Christian Sandvig, Kevin Hamilton, Karrie Karahalios, and Cedric Langbort. 2014. Auditing algorithms: Research methods for detecting discrimination on internet platforms. *In Data and Discrimination: Converting Critical Concerns into Productive: A preconference at the 64th Annual Meeting of the International Communication Association.* (2014).

[47] M. J. Sergot, F. Sadri, R. A. Kowalski, F. Kriwaczek, P. Hammond, and H. T. Cory. 1986. The British Nationality Act As a Logic Program. *Commun. ACM* 29, 5 (May 1986), 370–386. https://doi.org/10.1145/5689.5920

[48] Tom W. Smith, Michael Davern, Jeremy Freese, and Stephen Morgan. 2018. General Social Surveys, 1972-2018. gssdataexplorer.norc.org

[49] United States Department of Labor. Bureau of Labor Statistics. 2015. Consumer Expenditure Survey, 2013: Interview Survey and Detailed Expenditure Files. https://doi.org/10.3886/ICPSR36237.v2

[50] Daniel Trielli and Nicholas Diakopoulos. 2019. Search as News Curator: The Role of Google in Shaping Attention to News Information. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, Article Paper 453, 15 pages. https://doi.org/10.1145/3290605.3300683

[51] Amy Voida, Lynn Dombrowski, Gillian R. Hayes, and Melissa Mazmanian. 2014. Shared Values/Conflicting Logics: Working around e-Government Systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. Association for Computing Machinery, New York, NY, USA, 3583–3592. https://doi.org/10.1145/2556288.2556971

Received January 2020; accepted March 2020