End-User Audits: A System Empowering Communities to Lead Large-Scale Investigations of Harmful Algorithmic Behavior

MICHELLE S. LAM, Stanford University, USA
MITCHELL L. GORDON, Stanford University, USA
DANAË METAXA, University of Pennsylvania, USA
JEFFREY T. HANCOCK, Stanford University, USA
JAMES A. LANDAY, Stanford University, USA
MICHAEL S. BERNSTEIN, Stanford University, USA

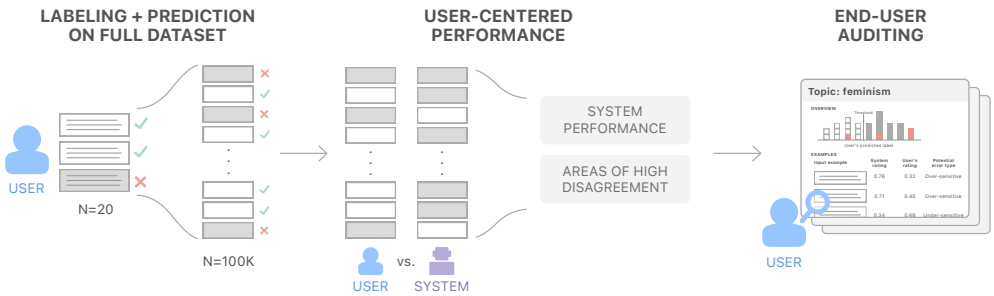

Fig. 1. Our end-user auditing approach: (1) An end user labels a small number of examples—our model uses data labeled by a diverse set of annotators on the same task to predict the user's labels on a much larger set of examples. (2) The predicted end-user labels allow us to calculate personalized metrics that estimate system performance for the user and flag areas of high disagreement between the user and the system. (3) Then, the user can lead an audit using an interface that visualizes system performance, highlights potential system errors, supports evidence-gathering, and assists with report authoring.

Because algorithm audits are conducted by technical experts, audits are necessarily limited to the hypotheses that experts think to test. End users hold the promise to expand this purview, as they inhabit spaces and witness algorithmic impacts that auditors do not. In pursuit of this goal, we propose *end-user audits*—system-scale audits led by non-technical users—and present an approach that scaffolds end users in hypothesis generation, evidence identification, and results communication. Today, performing a system-scale audit requires substantial user effort to label thousands of system outputs, so we introduce a collaborative filtering technique that leverages the algorithmic system's own disaggregated training data to project from a small number of end user labels onto the full test set. Our end-user auditing tool, IndieLabel, employs these predicted labels so that

Authors' addresses: Michelle S. Lam, mlam4@cs.stanford.edu, Stanford University, Stanford, California, USA; Mitchell L. Gordon, mgord@cs.stanford.edu, Stanford University, Stanford, California, USA; Danaë Metaxa, metaxa@seas.upenn.edu, University of Pennsylvania, Philadelphia, Pennsylvania, USA; Jeffrey T. Hancock, hancockj@stanford.edu, Stanford University, Stanford, California, USA; James A. Landay, landay@stanford.edu, Stanford University, Stanford, California, USA; Michael S. Bernstein, msb@cs.stanford.edu, Stanford University, Stanford, California, USA.

2573-0142/2022/11-ART512 $15.00
https://doi.org/10.1145/3555625

users can rapidly explore where their opinions diverge from the algorithmic system's outputs. By highlighting topic areas where the system is under-performing for the user and surfacing sets of likely error cases, the tool guides the user in authoring an audit report. In an evaluation of end-user audits on a popular comment toxicity model with 17 non-technical participants, participants both replicated issues that formal audits had previously identified and also raised previously underreported issues such as under-flagging on veiled forms of hate that perpetuate stigma and over-flagging of slurs that have been reclaimed by marginalized communities.

CCS Concepts: • **Human-centered computing → Human computer interaction (HCI)**; *Interactive systems and tools*; *Collaborative and social computing systems and tools.*

Additional Key Words and Phrases: algorithm auditing, algorithmic fairness, machine learning, human-centered AI, interactive visualization

ACM Reference Format:
Michelle S. Lam, Mitchell L. Gordon, Danaë Metaxa, Jeffrey T. Hancock, James A. Landay, and Michael S. Bernstein. 2022. End-User Audits: A System Empowering Communities to Lead Large-Scale Investigations of Harmful Algorithmic Behavior. *Proc. ACM Hum.-Comput. Interact.* 6, CSCW2, Article 512 (November 2022), 34 pages. https://doi.org/10.1145/3555625

Content Warning: This paper covers user-led audits of a content moderation system. Parts of this paper reference user-generated content containing offensive or hateful speech, profanity, and content pertaining to potentially triggering topics such as sexual assault.

1 INTRODUCTION

Algorithm audits have risen as powerful tools to hold algorithmic systems accountable. Leveraged in domains including healthcare, employment, housing, and criminal justice [2, 13, 55], algorithm audits have been defined as methods that systematically probe algorithms and observe their output to draw conclusions about their inner workings [50]. These audits have heightened technologists' awareness of the ways in which systems powered by machine learning (ML) can perpetuate harms, especially to marginalized groups [4].

However, one major limitation of auditing is its reliance on expert auditors: audits can only uncover issues where those auditors think to look. While algorithm auditors bring great value by advocating on behalf of end users, they are a small group of technical experts who cannot be expected to anticipate the full range of lived experiences that a technology will impact [16, 33, 43].

End users, on the other hand, do possess this rich situated knowledge of the particular impacts that algorithmic systems have on their own communities, and they already identify problematic behaviors of algorithmic systems through their everyday interactions with such systems [66]. Involving end users in the auditing process could help identify issues that auditors miss [19, 68, 74]. However, translating from a striking observation to a full audit often requires a systematic test of the algorithm's outputs.

In this paper, we pursue what we term *end-user algorithm audits*: system-scale audits led by individual, non-technical users. End-user audits could help marginalized groups bring attention to specific harms perpetuated by algorithmic systems and help development teams identify issues by engaging with stakeholders early on before they launch such systems.

It is currently very challenging for end users to engage in, let alone lead, algorithm audits. While finding a few examples of system errors can be illustrative, these do not constitute comprehensive audits, and system developers might disregard or have a hard time diagnosing systemic problems from a small number of specific examples [33]. Full systematic algorithm audits, meanwhile, require a significant amount of effort and expertise [4, 50, 63], which most end users do not have. Moreover, stakeholders and marginalized groups are likely underrepresented among expert auditors [16].

Our approach aims to make end-user auditing possible by projecting from a small amount of user input to predict a user's perspectives on a much larger set of comments—providing a launchpad from which they can investigate issues. Our insight is that the necessary information is already present in most machine learning datasets: labels from annotators in the training data. Typically, generalizing from a small number of labels from an end user would necessitate semi-supervised learning [69], but such methods struggle to learn complex concepts like "fairness" or "toxicity" that involve subjectivity. However, many of the existing datasets used to train algorithmic systems already contain a variety of perspectives and represent annotators with a range of identities. Drawing on techniques from recommender systems [62], we instead use a small number of labeled examples from the user (e.g., their ratings on the toxicity of a sample of text comments) to project their labels onto this space of diverse annotators; then, we utilize similar annotators' data to predict the user's labels for the full, much larger dataset (e.g., predicting the user's toxicity rating on hundreds of thousands of comments). This large-scale prediction enables users to investigate issues with the system—the target of their algorithm audit—quickly and with little overhead effort.

With estimated end-user labels in hand, we can allow users to enter the auditing process at a higher level to form and validate hypotheses about harmful system behavior. To guide users, we generate user-centered system performance metrics by treating the projected end-user labels as the correct labels and evaluating the target system's labels against them. We can then direct the user's attention where their expertise is most needed: areas where they disagree most strongly with the system. For example, consider a Black user auditing a hate speech detection system. After labeling a set of examples and seeing their labels estimated across the full dataset, the user may see that comments from other Black users discussing race have a high proportion of false positives—getting flagged incorrectly as hate speech. With our approach, users can formalize their hunches about system behavior in an audit report to credibly raise cases with system developers or the public.

To instantiate this approach, we introduce *IndieLabel*, a web-based tool that supports end-user algorithm auditing for comment toxicity models. These models are fundamental to algorithmic content moderation, the use of algorithms as opposed to human judgment to screen user-generated content on online platforms. This domain is particularly well-suited to end-user algorithm auditing because it is widely deployed, but faces frequent criticism for making severe user-facing errors [28, 44]. With *IndieLabel*, given as few as 20 labeled examples from a user, we can estimate their ratings on a dataset of over 100,000 examples. Users can provide labels, train their model, and start auditing in a matter of minutes. In the auditing portion of the tool, users can perform iterative rounds of hypothesis generation and evidence-gathering. Finally, their findings directly translate into audit reports designed to be shared with system developers.

In an evaluation with 17 non-technical users, we found that end users were able to successfully lead algorithm audits. Using the *IndieLabel* tool, they audited Perspective API, a popular comment toxicity model provided by Google [38]. Users identified a total of 76 issues spanning 57 distinct topic areas and uncovered previously underreported issues including over-flagging content about sensitive topics like race, slavery, and sexual assault; under-flagging on subtler forms of hate that perpetuate stereotypes and stigma; and over-flagging of several slurs that have been reclaimed by marginalized communities. Users were also independently able to uncover three main types of issues with the Perspective API that had been documented in prior research, supporting the correctness and viability of end-user auditing. During a half-hour auditing period, users conducted audits for multiple topics, and users found supporting evidence of potential issues in 75% of their audits. Further, users demonstrated a great deal of skill and creativity in their audits: users conducted audits on 36 unique topic areas that no other users in the study explored, and the vast majority of participants (all but one) felt that they had discovered issues that they hadn't anticipated beforehand.

These encouraging findings indicate that we can uncover important issues by enabling individual end users to lead algorithm audits. Crucially, more than simply crowdsourcing the work of algorithm auditing, this approach amplifies individual voices and allows individuals to take ownership over their findings. Demographic groups are not monolithic [9, 39] and recent work has highlighted the importance of acknowledging a plurality of opinions among users, since methods that average away disagreement among individuals may artificially inflate performance metrics and effectively silence minority voices whose opinions may matter most [23, 52]. We view our individual-oriented approach as a helpful complement to existing group-oriented auditing approaches.

In summary, our work makes the following contributions:

- **An end-user auditing approach**. We introduce a recommender system approach to model individual users' perspectives, which scaffolds the auditing process for non-technical users.
- **The *IndieLabel* system**. We introduce a web-based tool for end-user auditing. With this tool, users can provide a small number of labels, propagate their perspective to the full dataset using their personalized model, test their hypotheses at a large scale, and create an audit report to communicate their findings.
- **An end-user audit of the Perspective API**. We conduct a study with 17 non-technical users who use the *IndieLabel* tool to audit the Perspective API model. We find that users were successfully able to lead their own audits that yielded previously underreported insights on a host of potential system issues.

2 RELATED WORK

Here, we survey work across the HCI, ML, and algorithmic fairness communities on algorithm auditing and end-user engagement with algorithm design, which motivates end-user auditing.

2.1 Algorithm auditing

Algorithm auditing broadly describes a methodology used to investigate the workings of an algorithmic system to uncover harms such as bias, discrimination, or other problematic behavior [50, 63]. Given that algorithmic systems are often opaque and commercially owned, audit-based approaches have proven to be incredibly impactful, revealing harmful algorithmic behavior in domains such as housing [2, 21], employment [13, 14, 72], healthcare [55], search [49, 54, 61], and technologies such as facial recognition [9, 58] and automated speech recognition [40].

To design a *new* algorithm auditing approach, we must understand what differentiates algorithm audits from other system evaluation methods. In their survey of algorithm auditing, Metaxa et al. [50] define an algorithm audit as "a method of repeatedly querying an algorithm and observing its output to draw conclusions about the algorithm's opaque inner workings and possible external impact". They also highlight three features of audits that distinguish them from other kinds of algorithmic testing: (1) the focus of study: audits study the system itself, not "any particular component or a user's response to it"; (2) the scope of conclusions drawn: algorithm audits go beyond individual test cases to make broader declarations about the system as a whole; and (3) the position of the investigator: algorithm audits are generally external evaluations conducted with "varying levels of participation or consent from the entity being audited". We refer to this definition to benchmark the requirements for what an end-user audit must achieve.

2.2 The role of end users in algorithm audits

Algorithm audits are typically led by technical experts who have domain knowledge on algorithms or machine learning. These experts may be independent third-party auditors such as researchers and

data journalists, second-party external contractors who have been granted access to system code and documentation [72], or first-party employees with direct access to an algorithmic system [59].

While there is broad acknowledgement that stakeholder involvement should be a critical part of algorithm audits, such involvement is rare in practice [8, 16]. Historically, crowdsourced audits [63] have been the main algorithm audit strategy to involve users, but these audits involve users primarily as data contributors rather than as lead investigators. One line of crowdsourced audits involves passive data collection from opted-in users, for example, through browser extensions or automated scripts that gather data from users' machines [17, 30, 51, 61, 67]. Crowdsourced audits may also involve data collection from users who are assigned specific predetermined tasks designed by the auditor [5, 42, 45]. Another variety of crowdsourced audits uses data contributions from volunteers in a manner akin to bug reporting. Mozilla's RegretsReporter browser extension is one example: it allows users to voluntarily contribute examples of regrettable experiences on YouTube [47]. Another recent work introduces crowdsourced failure reports whereby users can describe machine learning failures in their own words so that developers can detect AI errors [10]. Similarly, Beat the Machine introduced game-like incentives to encourage users to brainstorm examples that would trigger "unknown unknown" errors in predictive models [3].

Such research shows that users can bring helpful insights when they are allowed greater involvement in audits, but there remains a lack of algorithm audit research that allows end users to fully lead their own audits. Recent work on "everyday algorithm auditing" was the first to theorize about user behaviors in this vein [66]. This recent research highlights the important role that everyday users can play in detecting problematic algorithmic behavior through their regular usage of systems. Given that cultural blind spots may prevent machine learning practitioners from addressing fairness issues [33, 43], audits led by users show promise to fill in critical gaps. However, facilitating these kinds of audits is challenging: users need assistance with search inspiration, sensemaking, and remediation processes as they seek to uncover algorithmic biases [19]. Another recent line of work shares our goal of providing end users with the ability to audit models but primarily focuses on inspection of individual inputs to a model, allowing users to interpret model behavior and revise their inputs accordingly in their everyday use of a platform [73].

In this paper, we present a system that enables users to perform their *own* audits, and we allow users to identify issues beyond what they might encounter in their normal platform use.

2.3 Algorithmic content moderation

In our work, we focus on algorithmic content moderation because it is widespread across online platforms, but is widely criticized for its errors [6, 46]. A core issue with algorithmic moderation is that language is complex and highly context-dependent [25], so systems can fail when, for instance, models incorrectly learn to associate markers of dialects like African American English with higher toxicity scores [64], or users adversarially mask harmful comments with minor rewordings [34].

Content moderation systems have had disproportionate negative impacts on marginalized communities, who are also commonly targeted by online harrassment [70]. Research has found that transgender users and Black users experience higher rates of content and account removals, especially on content directly related to their identity [28]. Black women in particular are often silenced via content moderation in what Marshall [44] calls "algorithmic misogynoir". This is a domain where audits led by end users—especially those who are often silenced—could be highly beneficial.

2.4 Amplifying marginalized voices

A major goal of our work is to intentionally gather and uplift the perspectives of individuals whose opinions are not typically captured in this space. Recent research highlights that in many social computing tasks, there is a substantial amount of disagreement among the population, and current

ML approaches tend to treat this disagreement as noise and mask it with averaging or majority voting [24, 52]. This is harmful because annotator disagreements often capture important conflicting perspectives in tasks such as hate speech and misinformation detection [36, 41]. Techniques that adopt the perspective of the majority may end up silencing the voices of minority groups [57].

In light of such population-level disagreement, we must move beyond one-size-fits-all ML approaches and acknowledge differing perspectives. In the context of algorithmic fairness analyses, research has noted that dominant single-axis approaches (accounting for a single demographic attribute such as gender or race) may be inadequate to measure and counter discrimination [32, 39]. Building off of Crenshaw's influential work on intersectionality [18], researchers in the fairness community have called for approaches that foreground intersectionality [4, 9, 16], as have researchers in HCI [65]. Universalist frameworks, in attempting to capture large populations, can unfortunately tend toward color-blindness, failing to address the inequalities and harms that are not shared equally and that accumulate for specific subgroups [7, 56]. Group-based fairness approaches tend to abstract away the complexities and hierarchies of identity (especially for categories like race that are socially constructed and contested) and tend to treat oppressed groups as interchangeable [29]. In line with these ideas, we aim to present a complement to the status quo of group-based algorithm audits led in a top-down manner by technical experts. End-user auditing provides an avenue for individuals to surface underrepresented perspectives as they audit algorithmic systems.

3 END USER ALGORITHM AUDITS

In this section, we describe how we design our end-user auditing approach to lower the effort threshold and enable non-technical users to lead their own algorithm audits.

3.1 Motivating scenario

To demonstrate our goals for end-user auditing, we'll walk through a motivating scenario. Cameron is an active member of an online forum related to graphic design. She is a Black woman and has authored a variety of resources on racial justice in the design community, but every time she's shared this kind of content, her posts have been taken down.[1]

Cameron is aware that this forum leverages the Perspective API to automatically delete potentially toxic content, so she tries out an end-user auditing tool, *IndieLabel*, to try to right these wrongs. She labels 20 examples on the tool and verifies that her personalized model is working well. *IndieLabel* then surfaces comment topic clusters (e.g., "Black/Blacks", "Whites/Caucasians", "Book/Read", "Hate/Hating") where it appears that Cameron disagrees most with the Perspective API system.

She examines a candidate audit report for the topic cluster entitled "Black/Blacks." The tool surfaces to her that when treating her predicted labels as the correct labels, the system's accuracy in this topic area is only 64%, and the system's most frequent error type is over-sensitivity (labeling comments as toxic when Cameron was projected to view them as non-toxic). Inspecting a visualization of comments sampled from this topic area, she indeed observes a large proportion of comments that were misclassified by the system as toxic. Surveying comments in this cluster, she agrees with *IndieLabel*'s projected labels, and she discovers that a large set of the misclassified comments included a hashtag that Black designers in her community have used to speak out about experiences with racism. Combined with her own experiences on the forum, she now has a strong hunch that the system is spuriously flagging content that mentions the identity term "Black" at all. She saves examples from this topic cluster as evidence to support this hypothesis.

[1]Content moderation experiences involving disproportionate content removals for Black individuals on content related to racial justice or describing racism have been documented by Haimson et al. [28]

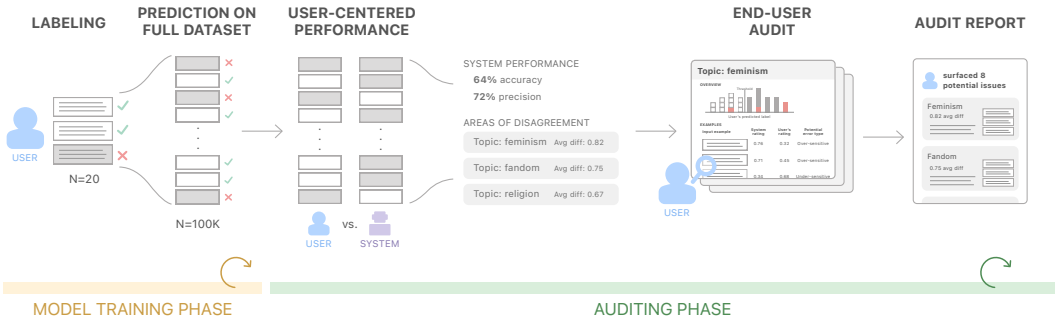

Fig. 2. An overview of our end-user auditing approach. In the *model training phase*, the user can provide a few labels to train their end-user model and iteratively tune the end-user model until it correctly predicts their perspectives. Then, in the *auditing phase*, our approach leverages the labels produced by the end-user model to scaffold the auditing process by proxying the user's labels on a large number of examples, enabling the user to identify areas where they disagree with the algorithm they are auditing.

After using the tool to perform keyword searches and fetch clusters of similar comments, she has a strong body of evidence that the system is over-sensitive to content related to racism in her community, and she writes a summary of what she's found. Cameron finalizes her audit report, and *IndieLabel* sends it to both the forum moderators and the Perspective API developers. The moderators review the report and realize that this is a major issue that they hadn't noticed. Inspired by Cameron's suggestions, they set up a manual rule that sends comments related to racial justice for human review, and they set up a streamlined appeal process so that incorrect flags can be reversed. The Perspective API team sees the report and agrees that this is undesirable behavior. They gather more examples of comments related to racial justice to improve their model, and they include the changes in their next model update release. Cameron sees these updates show up as follow-up items on the audit report and is encouraged by the progress. She revisits *IndieLabel* over time to investigate issues as the content moderation system evolves.

3.2 Approach

Our end-user auditing approach works toward lowering the high *effort* threshold of algorithm auditing by introducing a personalized model to scaffold the auditing process. This model allows us to label the full test set as the end user would, which then allows us to set up an audit that measures system performance from the vantage point of the user rather than that of the average annotator.

Our approach is broken into two main phases: (1) the *model training phase* and (2) the *auditing phase* (summarized in Figure 2). The purpose of the *model training phase* is to capture the user's perspectives in the auditing task domain so that the model can predict the user's labels on the full test set: i.e., if the user is auditing a comment toxicity detection system, we aim to capture the user's opinions on what constitutes a toxic comment. Then, the goal of the *auditing phase* is to leverage the personalized model to guide the user through the algorithm auditing process: i.e., assisting the user in identifying topic areas where the system seems to be flagging comments as toxic when the user views them as innocuous.

3.2.1 Dataset prerequisites. Our two dataset prerequisites are often already satisfied by modern machine learning datasets: we need to ensure that we have (1) enough data as would be needed to train an ML model, and (2) training labels per annotator (rather than just aggregate majority vote labels). The first requirement stands because the dataset must be sufficiently large to support

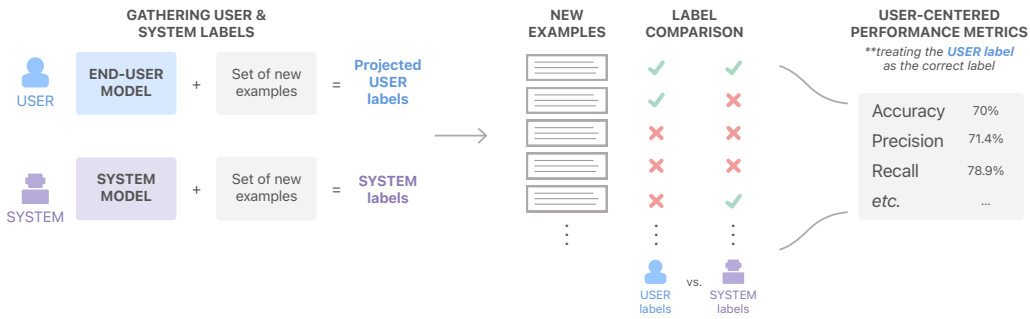

Fig. 3. With our end-user auditing approach, we can predict user labels and compare them to the system's labels to calculate user-centered versions of traditional model performance metrics, by treating the user's labels as the correct label and evaluating how well the system aligns with these labels.

personalized user model training and to provide a large enough body of evidence to understand system behavior in an auditing context. The second requirement stems from the recommender system approach that we take, which models the varied perspectives provided by prior labelers. Many datasets already gather this information along the way to generate ground truth evaluation datasets, but this information can also be generated fairly easily using a crowdsourcing platform such as Prolific.

3.2.2 Model training phase. Given such a dataset of input examples paired with multiple independent labels, the first phase of our auditing approach builds a personalized model for a new user. To train the model, we sample a small number of training set examples for the user to label. We used 20 examples in our work to demonstrate the feasibility of this method with minimal upfront user labeling effort (Section 5.3). Depending on the use case, various sampling strategies can be applied: for example, if the target system uses a threshold to separate classes, we can upsample close to this decision boundary to gain a better understanding in this critical region. We used a stratified sample based on the aggregate labels provided by the dataset annotators so that examples across the possible score range would be represented.

Then, we use a recommender system approach to predict the user's labels for all other comments in the dataset. Just as recommender systems can draw on correlations between users' ratings to predict the current user's unrated items, our system can label new examples as the user would based on their provided labels for a small sample and annotators' labels for the full dataset of examples. Recommender systems are a major domain of active research, and our present work does not intend to innovate new algorithms. We instead demonstrate that our approach succeeds even with a very simple model: the SVD algorithm [62]. This algorithm only requires a sparse matrix of user IDs, item IDs, and associated labels, and it outputs a full matrix that predicts a label for the new user on items they did not rate. Unlike modern deep learning recommender systems, which require significant time to retrain, SVD enables quick, interactive model training, so we can accept additional labels or modified labels and easily retrain our model. On additional rounds of labeling, we can leverage smarter sampling strategies like uncertainty sampling or upsample in areas where the user diverges most from the target system or other users. Since SVD is not a state of the art algorithm, this means that our results are a conservative estimate of how well this approach could work with more complex models.

3.2.3 Auditing phase. Once the user has sufficiently trained their model, we leverage this end user model to guide them through the algorithm auditing process. Without any model, the user would

have an unorganized set of examples to sift through. Even if they could cluster comments into topics, they would need to manually inspect hundreds of individual topic areas with no sense of where the system might be failing for them; within each topic area, they would need to pore over thousands of examples (effectively labeling each one) to find errors. Only after these labor-intensive tasks could they move on to identify higher-level trends in the system's behavior.

However, with our model as a starting point, we can triage users' attention by highlighting topic areas[2] where system failures appear most likely, and we can set up candidate audit reports for these topics. For many topics, there may not be a difference in opinion between the user and the system, but there will be certain topics where the end-user model has significantly different labels than the system; throughout the auditing phase, we focus the user's attention on such areas. For each topic area, we surface *user-centered performance metrics* that enable quick assessment of the types of errors that are occurring (Figure 3). For example, the user-centered F1 score would be calculated by treating the user's predicted labels as the correct labels; this stands in contrast to a typical F1 score that would be calculated using average labels from a set of annotators. Equipped with a tool that provides a user-centered notion of performance, users can review system behavior, gather evidence, and form hunches about where issues might lie in an iterative auditing interaction.

3.2.4 End-user audits in practice. Returning to the definition provided by Metaxa et al. [50], end-user audits fulfill the core requirements of an algorithm audit. The targets of our audits are algorithmic systems (ML models), the goal of the audit is to make claims about the system as a whole (reviewing system behavior on a comprehensive dataset), and the auditor might not have consent from the entity that is being audited (end users can use our auditing approach without internal access to the system).

We envision two ways for end-user audits to be adopted by real-world platforms. First, system developers could directly host the auditing tool on their platform; for example, Twitter could link to an end-user auditing portal from the Reporting menu on a tweet. In this case, they could directly use the model's training data (or some cleaned version suitable for public use) for the end-user auditing system's core dataset. If system developers host their own end-user audits, users could directly share their reports with the developers in a closed setting; this would bear similarities to a first- or second-party audit [59, 72]. This kind of direct integration with an existing platform may be ideal in many cases, as prior research has recommended the introduction of affordances that integrate bias detection and reporting directly into the platform itself [19].

Second, if a platform does not opt in to host end-user audits, a third-party could host their own auditing platform. For example, a non-profit working against harassment of journalists could host a site for end-user audits of Twitter. This third-party would need to scrape the target system and gather diverse annotator labels from a crowdsourcing platform to collect the core dataset. Our intended theory of change here would be for users to share their audit reports in a public forum (in a similar manner to everyday algorithm audits that achieved accountability by publicizing errors [66]). Similar to traditional external algorithm audits, our end-user audits would primarily seek to effect change by naming and drawing attention to problematic system behavior.

Just as with any external report or algorithm audit, we anticipate the possibility that adversarial actors may submit reports in bad faith, for example, reporting truly hateful comments as non-toxic. Thus, any system incorporating end-user audits should require system developers to perform due diligence and thoroughly review the evidence underlying audit reports before any mitigation steps are taken. Our approach requires auditors to share their full process from their input labels to the

[2]While we frame our audits around topic areas, there is flexibility around the particular clustering method used; our approach can be used with existing metadata, manual annotations, or automated clustering. We used Sentence Transformers [60] and BERTopic [27] to generate topic clusters in an unsupervised manner.

examples they viewed and saved as evidence to their written audit report, which allows developers and members of the public with whom the audit is shared to quickly judge if the provided labels are malicious or nonsensical. In line with the goals of traditional algorithm audits, the central goal of end-user audits is to expand system developers' awareness of potential issues that might arise from their system, and they must still consider the veracity of inbound reports.

3.3 The *IndieLabel* system

To instantiate our end-user auditing approach, we built the *IndieLabel*[3] web application that is designed to audit comment toxicity models. While specifically designed for this application area, the tool can be adapted to any problem area with text-based examples that are associated with a binary classification label and an underlying confidence score. Our dataset and system are available on a public Github repository.[4]

3.3.1 Problem area and dataset. For the *IndieLabel* system, we defined our problem setup a bit more narrowly to aid user understanding. First, we assumed that the target system—the system against which users will conduct algorithm audits—assigned binary labels (i.e., "toxic" or "non-toxic") via comparison to a threshold score. Second, we assumed that this classification task was directly tied to a real-world action (e.g., the deletion or downranking of a post). While our approach can be applied to regression tasks, we found that it is easier for non-technical users to understand binary classification applied toward concrete real-world outcomes. This problem setup is aligned with that of many real-world ML systems on platforms such as YouTube and Facebook [6, 11] and is directly applicable to a variety of language-based tasks (e.g., hate speech, misinformation, clickbait, spam detection) as well as vision tasks (e.g., facial recognition, object detection).

We selected the comment toxicity problem area because it is a high-impact, user-facing area that is widely disputed, and we selected the Perspective API[5] as our target system. Perspective is a well-known API provided by Jigsaw (a unit within Google) that uses machine learning to detect "toxic" comments. Given that the Perspective API is trusted by featured partners like The New York Times, Reddit, Disqus, and The Wall Street Journal, its toxicity predictions have significant real-world impact and are thus a valuable audit target.

Our core dataset was curated by Kumar et al. [41] and consists of toxicity labels from 17,280 participants for a set of 107,620 comments sourced from Twitter, Reddit, and 4chan. To generate this dataset, annotators rated comments on a 0–4 (5-point) Likert scale ranging from "Not at all toxic" to "Extremely toxic." We thus had users of our *IndieLabel* tool provide ratings along this same scale, and following Kumar et al., we adapted these ratings to a binary toxicity detection task by labeling comments with a score of "Moderately toxic" (a 2 on the scale) or above as "toxic" and those with a score below that as "non-toxic." We remapped the 0–1 Perspective API score-range to the 0–4 scale with a constant scale factor of four (our toxicity threshold at 2 thus mapped to a 0.5 on the original Perspective score-range, which is designed to be interpreted as a 50% likelihood of toxicity and aligned closely with Kumar et al.'s findings on a suitable Perspective threshold range).

We note that the term "toxic," while frequently used in the domain of algorithmic content moderation, is a vague term that is often used as a catch-all for behavior that platforms deem undesirable. Since we are auditing the Perspective API, we adopt this terminology, but targeted delineations of unwanted behavior (e.g., spam, harassment, incivility) are more useful and actionable.

[3]The name of our tool is a nod to independent record labels (or "indie labels") that capture the independent spirit of our end-user audits and reference the fact that these audits stem from an initial set of independent, user-provided labels.
[4]https://github.com/StanfordHCI/indie-label
[5]https://perspectiveapi.com/

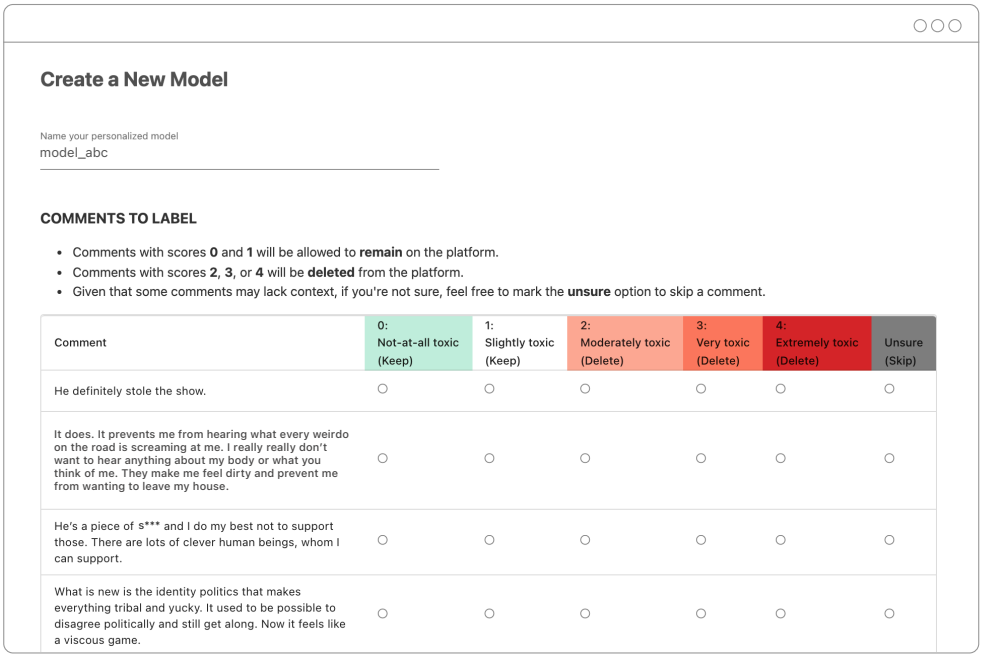

Fig. 4. The *Labeling* page of the *IndieLabel* tool. Here, users view a sample of comments in table form and provide their labels on a 0–4 score range. Users can select "unsure" to skip a comment and load a replacement.

3.3.2 System design. The *IndieLabel* system consists of two main interfaces: the *Labeling* page and the *Auditing* page.

First, the *Labeling* page (Figure 4) is the main hub for users to train and inspect the performance of their personalized model. As a reminder, our overarching task is binary classification via a threshold to a continuous score range. To solicit user perspectives on examples of varying potential toxicity, we provide a stratified sample across 5 score bins to cover the 0–4 score range: $[0.0, 0.5)$, $[0.5, 1.5)$, $[1.5, 2.5)$, $[2.5, 3.5)$, and $[3.5, 4.0]$. Comments in the training set were assigned to these score bins based on the median score provided by the prior dataset annotators. To gain a better understanding of users' decision boundaries, we upsample closer to the threshold: we sampled 2, 4, 8, 4, and 2 comments from each of the score bins, respectively. After they've provided labels and trained their model, users can view model performance on the validation set and may optionally fine-tune their model through additional labeling.

Next, users can proceed to the *Auditing* page (Figure 5). This interface is broken into two main sections: (1) the Audit Report Panel (A) and (2) the main Auditing Panel (B). The Audit Report Panel is a side-panel where users can view action items: areas where they should dive deeper on potential system errors. These action items are surfaced as a menu of "Unfinished Reports" (A7). Upon opening a candidate audit report, the user can see its core details: the topic area (A1) and error type (A2) that have been surfaced by the user's model. They can start tackling this report in the Auditing Panel.

The main Auditing Panel contains a series of visualizations to help the user to understand the behavior of the content moderation system and is where they will direct most of their attention. First, an *overview visualization* highlights topics for which the user's ratings are likely to deviate most from the system's (not pictured, but shares the design of the visualization in Figure 5(B3)). We visualize these topics in a histogram plot where each box represents one topic, and the x-axis

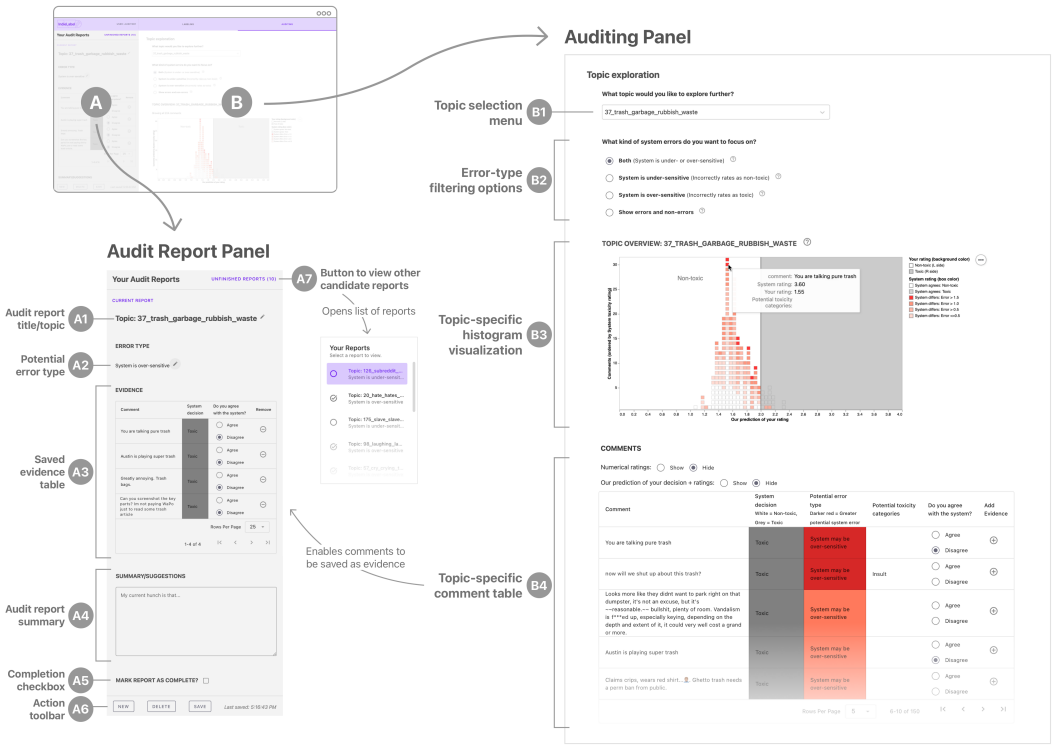

Fig. 5. The *Auditing* page of the *IndieLabel* tool. The Audit Report Panel (A) is a sidepanel that serves as a scratchpad for the current audit to keep track of the audit task/topic (A1, A2, A7), save evidence (A3), and summarize and submit a report (A4, A5). The Auditing Panel (B) is the main action area where users can make sense of system behavior across topics or within a topic of interest. Users can browse topics (B1), filter on different error types (B2), and view an interactive histogram visualization (B3) or comment table (B4).

represents the user's predicted toxicity rating on the 0–4 scale. The boxes are shaded according to the system's rating for the same topic; boxes are shaded in red where the system labeled the topic differently on average (with darker red indicating a greater magnitude of difference). Users can interactively hover over boxes to view details like the topic name, the system's average rating, and their own projected rating. They can click on a box to jump to that topic in a topic-specific exploration section below. This histogram visualization, inspired by that of ModelTracker [1], allows users to quickly identify topics with the highest potential error. At a glance, it gives users a sense of false positives, false negatives, and the relative spread of toxicity scores without requiring technical knowledge.

Next, there is a *topic-specific visualization* (B3) that highlights particular examples within the topic area where the user likely would disagree strongly with the system. This is visualized using the same method as the overview plot, but here each box represents a comment sampled from the topic cluster. The user can hover over boxes in the plot to view the comment, the system's rating, and the user's projected rating. Paired with the topic-specific visualization is a *comment table* (B4); this is a tabular representation of the same comments shown in the topic-specific visualization. Users can click on the topic visualization to jump to a comment in the table. In this table, users can see the comment, the system's decision (and underlying rating), and the potential error type of the system's decision (based on the user's predicted label). Users are also given a set of tools for

each comment; they can mark scratch notes for comments where they agree or disagree with the system, and they can click a button to port the comment over to the *evidence table* section A3 of the Audit Report Panel as they start to form an idea of potential error patterns for the topic area.

Below the comment table, there is a section for users to drill down further with keyword search or custom clustering. In the keyword search section, users can perform manual search for keywords or phrases (perhaps to explore ideas prompted by their topic-based exploration). In the custom clustering section, users can select seed comments and iteratively fetch and hone a cluster of neighbors (for example, to find more comments that display the same speech pattern). Both of these search modes return a comment table view where users can inspect examples and save evidence.

Once the user has formed an idea of the system's behavior, they can finalize their report. The final section of the Audit Report panel is a text-entry box for the *audit report summary* A4 where users can enter a summary of the evidence they've gathered, how they reached their conclusion, and suggestions to the developers on how to mitigate issues. We keep this as a free-form section so that users can express their opinions in an unstructured manner and provide insights that might not be clear through performance metrics or evidence alone.

3.3.3 Implementation. The system was implemented with the Svelte[6] frontend framework and a Flask[7] backend server, and it used the Altair[8] library for its data visualizations. We used the Surprise[9] library to train and evaluate our SVD model. We sampled comments from the dataset, users labeled these comments on the frontend, we ran the SVD model on the training set with these labels, and we retrieved predictions for the entire test set, which powered our frontend auditing interface and visualizations. All model training/evaluation and data processing took place directly on the Flask server, so the tool can be run locally or hosted on a remote server for wider deployment.

4 EVALUATION

Our study aims to evaluate the efficacy of the end-user auditing approach on a real-world task. In particular, we sought to determine whether our approach succeeds in lowering the effort threshold for non-technical users. This overarching goal broke down into two main research questions:

- RQ1—*Does modeling end users' labels enable end-user audits?* Does our approach model users' perspectives reasonably well? Do end users find the auditing process comprehensible?
- RQ2—*Do end-user audits yield insights?* What kinds of issues do users uncover in their audits? Do users discover unique insights?

Success in this vein would require that our modeling approach effectively aligns with users' labels and that non-technical users can make use of the information we surface based on our model. To address our research questions, we designed a user evaluation with non-technical users who would use our *IndieLabel* tool to audit the Perspective API for comment toxicity detection. Each hour-long study session consisted of two main phases: (1) a labeling phase where the participant provided ratings for a small sample of example comments and (2) an auditing phase where they completed three kinds of tasks, which we'll describe in greater detail in the next section. The study session was divided into 10 minutes for consent, onboarding, and a system tutorial; 10 minutes for the labeling phase; 5 minutes for a brief pre-audit survey questionnaire; 30 minutes for the auditing phase (10 minutes for each task); and 5 minutes for a post-audit survey questionnaire. During the onboarding phase, we provided a framing scenario for the auditing task (Appendix Section B).

[6]https://svelte.dev/
[7]https://flask.palletsprojects.com/en/2.0.x/
[8]https://altair-viz.github.io/
[9]http://surpriselib.com/

4.1 Auditing task design

Existing systems for algorithm auditing have a fundamentally different goal and task setup than ours: the goal of these traditional audits is to use data from a large number of users to uncover population-level trends in the behavior of an algorithmic system. In contrast, our end-user algorithm audit is an individual-level audit; its goal is to take on the perspective of a single user to uncover their personal insights on the behavior of an algorithmic system. Thus, our evaluation needed to assess how well our tool enabled *individual* users to investigate how well the content moderation system worked *for them*. We needed to select our baseline so that end-users were still the ones driving the audit (since they're the only ones who can conduct an audit from their perspective, and they're the only ones who can evaluate how well our tool helps them to do so). No prior tools existed that allowed end-users to drive the audit, so we set up our own baseline by having participants use alternative models to aid their auditing process.

To address RQ1, we used two model variants: (1) the *end-user model* that was trained using the user's ratings and (2) a *group-based model* that was trained using ratings from a single user-selected demographic group (users could select from one of five demographic axes—political affiliation, gender, race, LGBTQ+ identity, or importance of religion—to set up their group-based model). This group-based model baseline would allow us to understand how an end-user model provides value over and above what a group-based model, which doesn't require any labels from the end user, could provide. We also trained a *uniform model* based on ratings averaged across all annotators (solely for use in our technical evaluation, which we describe in Section 5.3).

We developed two kinds of auditing tasks: (1) a *fixed* audit where a user would audit from among a set of topics surfaced by their model and (2) a *free-form* audit where a user could perform an audit for whichever topics they'd like to investigate based on their own interests. For the *fixed* task, after each participant had trained their model, we generated a set of ten topics that displayed the highest proportion of potential system errors (false positives or false negatives based on end-user labels). Users were asked to conduct audits strictly from among this personalized set of topics. The fixed task allowed us to understand how well users could conduct audits when there was a clear topic and potential error type. Meanwhile, the *free-form* task allowed us to observe how users could take more agency over the audit process to explore potential issues that were most relevant to their own experiences. For this task, we asked users to conduct audits stemming from content moderation issues relevant to a community to which they belonged. Thus, this second task was more aligned with our ultimate vision of user-led audits that stem from users' unique perspectives. To help users to brainstorm for the free-form audit, we provided some question prompts (Appendix section A).

We thus had three final auditing tasks: (1) *Fixed audit, End-user model*, (2) *Fixed audit, Group-based model*, and (3) *Free-form audit, End-user model*. We only evaluated the group-based model for the fixed audit task since that provided a more consistent format for comparison between the model variants and between study participants. Given that the free-form audit was meant to observe how users express their own individuality in their audits, we were most interested in how users were able to leverage their end-user model for this task. We randomized the order of the fixed audit tasks to control for learning effects; the free-form audit was always kept as the final task so that users were sufficiently familiar with the tool to perform their own explorations.

4.2 Pre- and post-audit surveys

Participants completed a brief survey before and after their audit. The pre-audit survey asked general questions related to their perceptions of content moderation systems, their opinions on what constitutes a "toxic" comment, and their confidence in their ability to perform an audit. The post-audit survey covered (a) their views on the content moderation system, (b) their views on the

IndieLabel auditing tool, and (c) their views on the end-user versus group-based model. Most of these questions used a 7-point Likert scale and some questions had room for a brief short-answer text response. The full survey is included in Appendix section C.

4.3 Qualitative analysis

Both the pre- and post-audit surveys and the end-user audit reports had open-response fields to analyze. Here, our goal was to summarize the high-level themes that emerged from our participants, so the codes were our process, not our product [48]. The first author conducted an inductive analysis to summarize our participants' end-user auditing outcomes. The first step of this process was to read through all survey responses and audit reports multiple times. Then, the qualitative open coding [12] process was iterative and took place in two phases: in the first phase, responses were coded line-by-line to closely reflect the original data (e.g., "profanity used for emphasis," "dehumanizing word," or "survivors sharing experiences"). In the second phase, the codes from the first phase were synthesized into higher level themes (e.g., "types of harms" or "proposed solutions"). After these themes were generated, examples were coded according to the themes to characterize the different kinds of audit results that participants achieved with our system.

4.4 Participant recruitment

We sought participants who had little technical background. We recruited participants from university mailing lists, asking participants in a signup survey to indicate their level of understanding about algorithms and machine learning. We selected from among participants who either labeled "not at all" or "only a little" (the lowest on a 5-point Likert scale) for both of these topics.

We also sought to include a diverse set of perspectives. Building on the demographic attributes collected in our labeled dataset, we chose a subset of attributes that prior research found to be most influential on user perspectives of comment toxicity: gender, race, LGBTQ+ identification, importance of religion, and political affiliation [41]. Our signup survey included questions where users could self-identify along these demographic axes. In total, 17 users participated in our study. We had 10 women, 4 men, and 3 non-binary participants; we had 6 Asian, 4 Hispanic, 2 White, 1 Black, and 4 multi-racial participants (Asian/Hispanic, Asian/White, Hispanic/White, and Black/White); we had 6 participants who identified as members of the LGBTQ+ community; for importance of religion, we had 7 who responded "not important," 5 who responded "not too important," and 5 who responded "very important"; we had 10 liberal, 3 independent, 1 conservative, 3 other self-specified political affiliations (1 socialist, 1 leftist, and 1 who didn't want to identify with provided categories); we had 15 participants aged 18-24, 1 aged 25-34, and 1 aged 35-44. We compensated our participants with a $25 Amazon gift card for completing the hour-long study session. All studies were conducted remotely over video conferencing due to the ongoing COVID-19 pandemic.

5 RESULTS

We found in our study that non-technical users were successfully able to lead algorithm audits. Users identified potential system errors in the vast majority of their audits, and they explored a broad range of unique topics and issues—replicating and going beyond issues found in traditional audits of Perspective API. End-user modeling helped users discover issues beyond what group-based modeling could reveal, and users felt that *IndieLabel* helped them to uncover unanticipated issues.

5.1 Overview

Using our tool, end users were able to conduct their own algorithm audits and discover potential issues with the Perspective API-based content moderation system. Users on average audited for 5-6 topics during their study session and completed a comparable number of audits across the three

Fixed audit, Group-based model (7)	Fixed audit, End-user model (10)	Free-form audit, End-user model (19)
africa/africans, f*g**t/f*g**ts, furthermore/citadeling, nigeria/nigerian, smell/smells, stranger/strangerthings3, wwe/raw	anime/manga, boobs/tits, dream/nightmares, happy/happiness, mexicans/mexico, pay/money, punched/fight, scam/scammers, sugar/sugardaddy, thank/thanks	asian/asians, black/african, cat/cats, china/chinese, die/death, eat/eating, fandom/fandoms, hes/guy, jews/jew, rape/raped, religion/religions, teacher/teachers, wtf/happening, Keyword search: b***h, Keyword search: feminine, Keyword search: latino, immigrant, Keyword search: le$, Keyword search: queer, Multiple keyword searches for replacements of flagged words (ex: k!ll, h0e, r4pe)

Table 1. Summary of *unique* audit topics (investigated by only one user). Users explored the greatest number of unique topics in conditions where they used their end-user model, especially in the free-form audit.

tasks. Across all study sessions, there were 101 completed audits; participants explored 57 *distinct* topics among these audits. Out of these 57 distinct topics, 21 were investigated by multiple study participants, and 36 were investigated by just one participant, which points toward the value of enabling individuals to lead their own audits. The audits were often fruitful: users found evidence of an issue in 75% of their audits. However, the error type predicted by the model in candidate audit reports wasn't always the same issue that the user ended up raising: users confirmed the same error type surfaced by their model in 51% of fixed audits. Users included an average of 6.3 pieces of evidence per audit and wrote on average 45.9 words for each audit summary. Given the 30-minute auditing phase time limit, we note that participants were given enough time to explore topic-specific system behavior and gather evidence for one round of hypotheses, but they did not perform the additional hypothesis iteration, interactive system probing, statistical testing, and formal audit publishing that are normally a part of a full audit.

5.1.1 Issues identified. Overall, auditing Perspective API with a threshold at 50% toxicity likelihood, the main errors that participants identified were false positives, where Perspective rated comments as toxic that users thought were non-toxic. Of the 76 audits that found evidence of an issue, 68 identified an issue of system over-sensitivity (too many false positives), 4 identified an issue of system under-sensitivity (too many false negatives), and 4 identified an issue with both over- and under-sensitivity within a topic area. Our qualitative analysis, covered in Sections 5.2.1–5.2.3, reveals that users identified a broad set of error types along with justifications and suggested fixes.

5.1.2 Fixed vs. free-form audits. In the *fixed* auditing task, users uncovered slightly more unique system issues with the aid of the end-user model than with the group-based model. Using their group-based model, participants explored 21 distinct topics, and 7 of those topics (33%) were investigated by just one participant. Meanwhile, using their end-user model, participants explored 24 distinct topics, and 10 of those topics (42%) were investigated by just one participant. Table 1 summarizes the unique topics explored in each study task. Furthermore, users were more effective at discovering system issues with the help of the end-user model: users verified issues in 76% of audits with their end-user model compared to 67% of audits with their group-based model. The *free-form* auditing task demonstrated that users were able to conduct creative and unique audits that stemmed from their own interests and experiences. In this task, users were slightly more effective than in the fixed task; they found potential system issues in 79% of their audits.

Trend	Subcategory	Explanation/Example
Complicated keywords with multiple meanings	Profanity (f***, a**, s***)	Profanity is used in multifarious ways; while it can be used to target others, it is often used for functions like expressing heightened emotion, adding emphasis (often in a positive direction), acting as a filler word, or adding humor.
	Negative-valence words (hate, shut up)	A word like "hate" also is used more casually online, so it can be used to refer to negative comments that one might receive, to express actual ill-will towards someone else, to humorously complain about something, or to express dislike for something.
	Name-calling words (coward, ignorant, idiot)	Terms like "coward," "ignorant," "shut up," and "idiot" all can be used negatively, but often are not, and the system tends to be over-sensitive to usages of these terms.
Reclaimed slurs	Terms like b***h, n***a, gay, and queer can have neutral or positive meanings within a community	Participants noted the various positive interpretations of these terms and emphasized that context about the poster and audience of the comment was necessary to determine whether the slurs were acceptable. Over-moderation of these terms was viewed as problematic because it would disproportionately impact members of these communities who are already marginalized.
Types of harms to consider	Insults or attacks	"Many outsiders use personal attacks and generalize the group." (P1), "The moderation of the word mostly comes from insults" (P6)
	Universally offensive slurs	"[Words like the r-word] are generally seen as universal insults and often ableist language that, regardless of usage, are toxic." (P8) "This is a very offensive slur against disabled people and the autistic community" (P2)
	Stereotyped messaging	"It insinuates that immigrants from a specific place are worse than from a different one" (P10), "['Gay' is used] in a way that continues to stigmatise the word and community" (P3), "Most comments are toxic because they assert whites as the superior race" (P13)
	Erasure or silencing	"While flagging [n***a] is probably because of trying to catch racist speech, it seems to be ignoring Black history" (P3), "Individuals speaking about their experiences could have their content falsely blocked in a shadow-banning sense" (P8)

Table 2. Summary of converging trends raised by study participants

5.2 Characterizing end-user audit outcomes

In our qualitative analysis of end-user audits, we found that users converged upon several clear sets of issues, expressed valid diverging opinions about system behavior on divisive topics, and surfaced unique insights that other users had not uncovered.

5.2.1 Convergence among flagged issues and suggested fixes. Several trends emerged among the issues and fixes raised by end-user audits, which we summarize in Table 2. Participants noted that certain keywords were challenging to moderate because they have multiple, vastly differing meanings, but the system tended to rule based on the toxic interpretation of the keyword. Many participants also raised that certain slurs have been reclaimed by communities and are acceptable when used among community members, but were over-flagged by the system. Users also converged on several types of harms that they cited as most problematic. While the first two harms (insults/attacks and offensive slurs) are encapsulated by Perspective API's available attributes,[10] the last two (stereotyped messaging and erasure/silencing) may not be fully captured by the API and present potential areas for further development. In addition to surfacing system errors, users offered potential solutions based on their experience (see Appendix section D.1).

5.2.2 Valuable divergence among issues and justifications. When users audited the same topic, the end-user auditing approach was able to solicit insightful, *differing* perspectives on whether issues existed, why certain system behavior was problematic, and what solution strategies could work. These results demonstrated the power of our approach to shift from monolithic auditing perspectives to a paradigm where audit results could come from different populations who might see things differently. There were 16 topics that were independently investigated by multiple study

[10]Perspective API provides toxicity, severe toxicity, identity attack, insult, profanity, and threat attributes.

"b***h/b***hes" (N=6)	#	"r*t**ded/r*t**d" (N=5)	#	"whites/minority" (N=5)	#	"n***a/n***as" (N=4)	#
Over-sensitive	6	No error found	5	Over-sensitive	3	Over-sensitive	3
System should consider whether used as an attack	6	Term is used as an insult	3	Under-sensitive	1	System should consider that slurs can be re-claimed	2
System should consider gender identity of poster	2	Comments promote an ableist attitude	2	Race-related speech should not be censored	2	Filtering the n-word unfairly impacts the Black community	2
System should consider whether targeting at one woman or all women	1	Term is a universally offensive slur	2	Comments contain implicit assumption of whites as superior race	1	System should not moderate usage of the n-word with an -a ending	1

Table 3. Summary of points raised by participants for topics that were investigated by 4 or more individuals. Participants were able to surface multiple valid perspectives even in the same audit topic area.

participants (see Figure 9 in the Appendix); we further analyzed four of these topics that were investigated by four or more participants.

Even with this relatively small number of overlapping participants, there were notable differences in what insights they provided (Table 3). Some participants disagreed on whether errors existed within the "whites/minority" and "n***a/n***as" topic areas. Even when participants agreed on the presence of an error, as in the b-word and r-word topic areas, they often provided different explanations for what made this behavior problematic or had different suggestions on how to address the error. For example, even though all users felt that the system correctly flagged examples as toxic in the r-word topic area, some users took issue that the slur was invoked as an insult against others while other users raised a higher-level issue that the term itself was universally offensive since it promoted ableist attitudes. In the "whites/minority" category, there was disagreement about whether comments targeted towards white users should be flagged as toxic; some participants felt that there needed to be space for race-related speech (for example, to criticize white people's position of power in society) while other participants felt that the comments directed at "whites" were rightly flagged. These results are consistent with prior work highlighting disagreement among users in domains like comment toxicity [24, 41]. Our findings suggest that end-user audits provide value by soliciting a plurality of opinions on problematic content, which can lend developers a richer understanding of the trade-offs surrounding a system error.

5.2.3 Unique user insights. Going beyond converging and diverging insights, we found that users were able to uncover system issues that had not received much attention previous to our work. Table 4 highlights several case studies with notable insights. The major trends we identified among these unique user findings were: (1) Over-flagging on comments containing marginalized group identity terms, (2) Under-flagging on subtler, implicit hate speech, (3) Over-flagging on content sharing personal experiences around difficult topics, and (4) Missing nuance and awareness of consent in body-related discussion.

5.2.4 External validity: comparison with past expert-led audits of the Perspective API. To investigate the external validity of the issues uncovered by these user-led audits, we briefly summarize issues that have been documented by previous expert-led audits of Perspective API. Researchers have found that adversarial approaches—such as word changes (e.g., misspellings), word-boundary changes, and the addition of innocuous words (e.g., "love")—effectively fool the Perspective model [26, 34]. Another major documented issue is that the Perspective model displays *false positive bias* for AAE (African American English) dialect examples [64] and for a host of demographic identity terms that were over-represented among the toxic comments in the training dataset [20]. Prior work has also raised that the system is highly sensitive to the presence of keywords, especially profanity [34, 35],

Trend	Description	Example case study
Speech targeting marginalized groups	Over-flagging was harmful. Subtler, veiled harms were not detected.	*Latinx users, immigrants.* P10 found that many comments were flagged simply for containing the word "Latino" even in the absence of profanity or other apparent markers of toxicity; they noted that "talking about people's identity is important and shouldn't instantly get flagged." They also highlighted examples of veiled hate that were not flagged by the system.
		Religious users, Christians. P12 expressed that over-flagging was problematic and that it "[is] fair to express freedom of speech, in term of one's opinion on religion."
		LGBTQ+ users. P3 surfaced that even when the word "gay" is not used in a personal attack, it is often used in a derogatory manner that "continues to stigmatize the word and community."
Sharing personal experiences around difficult topics	Over-flagging resulted in feelings of censoring or silencing.	*Race-related comments.* P13 raised an issue that the system was flagging comments discussing race even when they didn't contain slurs. They suggested that "the system needs to be aware of racial dynamics that need to make space for white people to be criticized as they have occupied a high level of power."
		Slaves, slavery. P12 found that the system was over-sensitive in this area, marking such comments as toxic "even though slavery occurred for hundreds of years in the U.S. and still occurs today." This participant cautioned that "this isn't a topic that should be swept under the rug."
		Rape, sexual assault. P14 brought up that "social media can be an outlet for survivors to talk about their experiences, for news channels to share information regarding sexual assault cases, and for people to stand in solidarity for victims. By deleting all occurrences of the word rape, we are acting as if it doesn't exist and isn't a problem."
Body-related discussion	Discourse related to the body involves more nuance. Content moderation should account for consent.	*Fat, obesity.* P1 conducted an audit for a topic on fat and obesity and surfaced that "words pertaining to weight or body and 'fat' are not always negative. [They] can sometimes be in support of someone."
		Explicit references to human anatomy. P14 expressed that the system was over-sensitive and that natural anatomy-related words (e.g., "p*ssy") can be used in colloquial, non-profane ways. P13 felt that explicit comments should be moderated, but the reason they provided was that many of these comments were targeted at the original poster without consent.
Community comparisons	Some communities faced differing content moderation behavior.	*LGBTQ+ community.* Comparing comments related to Gay users to those related to Trans users, P2 noted similar trends of incorrect flagging for within-group usages of reclaimed slurs.
		Black/African vs. Jewish community. P7 found that the system appeared to be over-sensitive to comments containing terms like "Black" and "African", but the system appeared to be under-sensitive for comments containing terms like "Jew" and "Jewish."
Miscellaneous findings	Participants uncovered a variety of other unexpected system issues.	*Fandom, fan culture.* P6 found that the system appeared to be over-sensitive to content related to fan culture even in the absence of hate speech or profanity.
		Adversarial use of symbols and misspelling. P8 and P13 looked into misspellings like "k!ll" and "wh0re" that may be used to evade flagging.
		Belligerent words. P4 reported that "words like fight, punch, or kick were often flagged "even when they [were] used metaphorically or for emphasis."
		Smell-related words. P14 noted that the system appeared to be over-sensitive to comments containing words like "smell" or "smelly," but that these word could be harmful if used to "generalize or insult a people group."

Table 4. Highlighted examples of unique insights raised by study participants

and that profanity in particular is a significant factor in driving misclassification of AAE [31]. As we detailed in prior sections, end-user auditors in our study independently surfaced all three of these issue types—adversarial inputs that obfuscate commonly-toxic terms (Table 4), false positive bias for demographic identity terms (Tables 2 and 4), and system sensitivity to profanity (Table 2)—as well as additional issues that were rarely reported in prior Perspective API audits.

5.2.5 Failure modes of end-user audits. While our end-user audits successfully identified a number of system issues, we observed several failure modes. First, sometimes our end-user and group-based models did not accurately capture users' views ("[both models] were unfortunately a bit inaccurate

in capturing my views" (P5), "I think neither [model] captured my views" (P4)). In these cases, users had to perform more manual labor to inspect comments since they could not rely on their model to correctly label on their behalf, so they made slower progress in conducting audits.

Next, some participants did not conduct very thorough audits. While in some cases a clear trend in a topic area is apparent after viewing a small number of examples, this kind of approach (which in the worst case defaults to one-off bug reporting) could lead to spurious audit results if intending to make claims about system-scale behavior. Several participants indicated that they had difficulties interpreting the histogram plot, so they stuck to the comment table ("I did not really understand the graph so I didn't use it" (P3)). Overreliance on this view could lead towards greater fixation on particular examples without an understanding of overall trends. Other participants did spend time to more deeply understand overall system behavior, but only added a small number of examples as evidence or only wrote a terse or vague summary (e.g., "The context is key" (P15)). While this behavior may stem from the time limits of our study, another issue may be that users find it difficult to articulate their views and could benefit from additional scaffolding on report authoring.

5.3 Technical evaluation

Our technical evaluation assesses whether gathering only a few labels from end-users provides adequate information to predict their labels on the test set. We found that our end-user model sufficiently captured users' perspectives and achieved better alignment than a demographic group-based model or a uniform model. In our previously described evaluation, participants provided ratings for an independently drawn sample of 40 comments (sampled using the same stratified sampling method described previously). Of these comments, 20 were used to train the user's personalized model (Figure 7), and 20 were held out as the participant's test set.

For each user, we evaluated the performance of five different models on this test set. First, there was the *end-user* model: this was the personalized model that was trained on the user's 20 training-set labels and used in *IndieLabel*. Second, there was the *group-based* model based on a single user-selected demographic group. This model emulated a group-based audit measuring system performance for a demographic group (e.g., Latinx-identifying users). To train this model, we sampled 20 comments labeled by annotators who matched the user's selected demographic group (using the same stratified sampling method). Then, for each comment, the label was the median rating among annotators from the selected demographic group. Third, there was the *uniform* model: a single aggregate model based on averaged ratings. This model emulated what a simple majority-rule system would decide, which is a common strategy used to determine "ground truth" labels for training datasets. To train this model, we again sampled 20 comments using the same stratified sampling method, but here the label was the median rating among all annotators.

To explore further improvements on personalization, we trained two deep learning recommender system models based on Gordon et al. [23]'s Jury Learning architecture. Both versions were trained with the users' 20 training examples, but the *group (DL)* version only used the user's demographic information while the *end-user (DL)* version made personalized predictions for the user (Appendix Section E). Unlike the SVD models used in *IndieLabel*, the DL models cannot train at interactive speeds, though future advances may make this possible. With these five model variants prepared for each study participant, we evaluated their respective performance on 20 participant-labeled test set ratings. For each model, we report the performance averaged across five trials with random train/test splits.

Comparing the raw 0–4 scale ratings, we found that the end-user model achieved lower MAE (mean absolute error) compared to the uniform and group-based models (Figure 6). We selected MAE as our primary performance metric due to its interpretability, as its units match the units of the toxicity score range, and changes to MAE are linear. However, we observed the same results

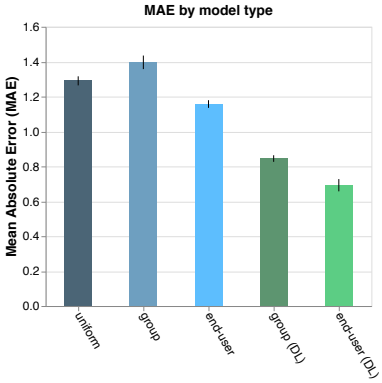

Fig. 6. Mean absolute error on participants' test sets (on 0–4 score range). The two end-user model variants achieve lower MAE compared to their corresponding group-based models.

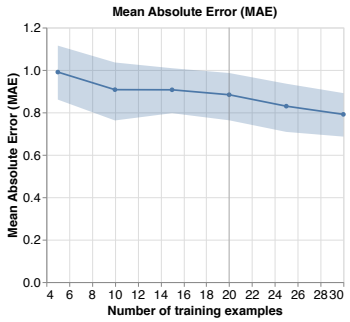

Fig. 7. Learning curve for participants' SVD end-user models (5 trials per datapoint). Additional labels beyond our choice of 20 don't substantially increase performance.

for MSE, RMSE, and classification accuracy, which we report in Appendix Section F. Among the SVD models, the end-user model achieved 1.16 MAE, SD=0.03 (group-based: 1.40 MAE, SD=0.03; uniform: 1.29 MAE, SD=0.05). For the DL models, the end-user model achieved 0.87 MAE, SD=0.23 (group-based: 0.85 MAE, SD=0.01). Noting the very large variance for the DL end-user model, we observed that two of the five randomly-initialized training trials likely reached a local minima, achieving a mean of 1.12 MAE, SD=0.02. Of the three that did converge, we saw a mean of 0.70 MAE with a small SD=0.009. Thus, we excluded these failed trials from Figure 6 and our subsequent comparison of the five model variants since in real-world deployments, model developers would be able to select a model that has properly converged.

Using a linear mixed-effects model with a fixed effect of model type and random effects of participant ID and trial ID (`MAE ~ 1 + model_type + (1 | participant_id) + (1 | trial_id)`), we observed a significant main effect of model type ($F(4, 266.08) = 113.15, p < .001$). A posthoc pairwise Tukey test found statistically significant ($p < .05$) differences between the mean MAE values of all pairs of models. Of particular note, the performance improvement of both end-user model variants over their corresponding group-based models was significant ($p < .001$ for both the SVD and DL models), which indicated that personalization using twenty end-user labels led to improved performance: in the case of the deep learning models, for example, a substantial improvement from .85 to .70 MAE.

5.4 User perceptions of end-user audits

Additionally, we wanted to understand our participants' perceptions of the end-user auditing experience. Participants found their auditing experience to be fruitful: all but one participant felt that they had discovered potential issues with the content moderation system that they hadn't anticipated, and only two participants felt that they might have discovered these same issues *without* a tool like IndieLabel. All 17 participants found that the tool was between somewhat useful and very useful in aiding their understanding of the overall, system-wide behavior of a content moderation system. Users had not anticipated prior to the study that they could lead such audits; in the pre-audit survey, only 35% of participants felt somewhat confident or confident that they would be able to perform an audit. However, in the post-audit survey, 77% of participants expressed that they felt somewhat comfortable, comfortable, or very comfortable with auditing the content moderation system (Figure 8).

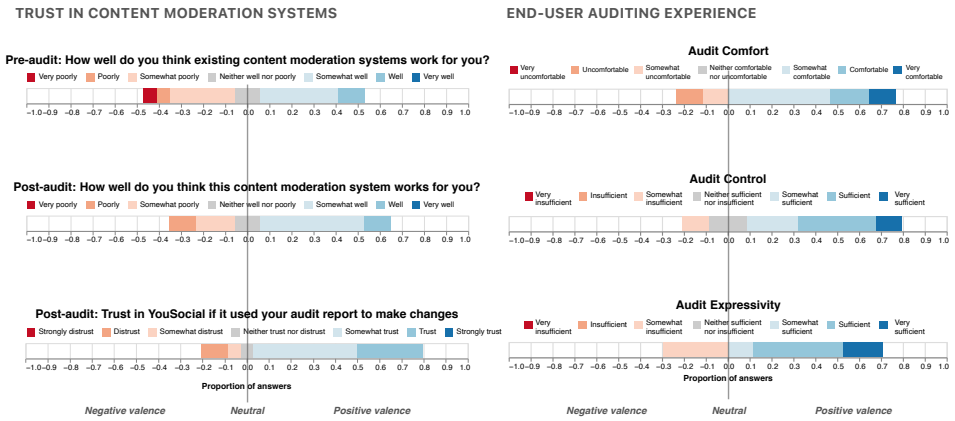

Fig. 8. Audits appeared to have some positive impact on users' trust in the target system (left). Participants indicated relatively high levels of comfort, control, and expressivity in their audits (right).

The auditing process also had some positive impact on users' trust in the target system. Before using the *IndieLabel* tool, 47% of participants felt that content moderation systems worked somewhat well or well for them, but this figure increased to 59% after participants used the tool (Figure 8). Platforms may find it beneficial to incorporate end-user auditing into their workflows to improve user trust: 77% of users felt that if they knew that a platform would be using their audit report to make changes, they would somewhat trust or trust its content moderation system.

Participants found value in both the end-user and group-based models: 82% felt that the end-user model was somewhat helpful, helpful, or very helpful in focusing their attention on potential issues that they found important; 77% of participants felt the same for the group-based model. When asked whether they would prefer to have only an end-user model, only a group-based model, both, or neither, all but 2 participants wished to have both models (1 participant wanted only an end-user model and 1 wanted only a group-based model). In open-ended responses comparing the models, 9 participants expressed that the end-user model seemed to better capture their views, 5 felt that both models worked similarly well, and 3 felt that neither system captured their views very well. When explaining why they felt that the end-user model captured their views, users said that it surfaced "keywords that may be used against communities with which I feel affinity" (P9) and that it "better captured my views and in particular, uses of certain language" (P6).

6 DISCUSSION

We have outlined an end-user auditing approach and have demonstrated that by modeling end users' labels, we can successfully reduce the effort barrier and yield useful audit insights. Here, we discuss opportunities, limitations, and areas for future research on end-user audits.

6.1 Opportunities unlocked by end-user auditing

6.1.1 Considering unique, diverging, and converging perspectives. As highlighted by our evaluation, a benefit of end-user-led audits is that they can surface unexpected, *unique* issues: issues that perhaps only a small number of users would discover and that may be more deeply buried among the "unknown unknown" issues of a system. Why should developers care about issues that impact a smaller number of users? The size of a group is not always commensurate with the severity of harms it experiences, so developers need to weigh and address acute harms that may impact minority identities. Another benefit of these audits is that when multiple users conduct audits for the same topic, we can build an understanding of the valid, *diverging* perspectives users hold. By

soliciting a plurality of opinions, developers may be able to better map out the trade-offs of their decisions in a certain topic area. This might enable them to make more explicit choices about whose voices and whose harms they will prioritize in their ML model in light of disagreement [23]. Lastly, end-user audits also provide useful signal when users express *converging* opinions on system errors. In situations where users are largely aligned in their views, the case for mitigation is clear, which can help to combat the struggles that system developers often face in deciding where to focus their efforts and how to trade off against various fairness concerns [33].

6.1.2 Building off of users' contextual expertise to develop solutions. By engaging in the auditing process, users can leverage their lived experiences to provide a richer understanding of the *reasons* underlying system errors and help design *potential solutions* that stem from their experience in a community [66]. When fixing system behavior in contentious topic areas, it is important to understand not only *whether* problematic system behavior occurs, but *why* users find the behavior to be problematic. Users may agree that behavior is problematic, but disagree on why (e.g., one user may find the r-word problematic because it's used to insult someone; another user may say that the r-word is universally unacceptable)—differing justifications could imply very different solution strategies. Meanwhile, users may disagree on whether an instance of behavior is problematic, but share similar guiding principles in general (e.g., users may disagree on whether the b-word should be allowed, but may agree that users within a community should be able to decide via a participatory process)—differing low-level judgements may distract from broader approaches that users may agree upon. End-user audits can provide a richer understanding of the thought process behind a user's judgment and allow developers to make decisions that more closely align with the user's perspectives.

6.1.3 Shifting the timing by which issues are surfaced. Additionally, end-user audits can provide an opportunity for system developers to catch issues from users *preemptively* before launch. While currently users cannot probe algorithms until they are already deployed, end-user audits can be set up as soon as the developer has a dataset and a version of their system to test. Then, rather than catching problematic behavior once a system has already caused widespread real-world harm, developers can hopefully catch issues early on via end-user auditors. Also, while formal audits are significant endeavors that can take months to complete, end-user audits are a lightweight option. Though end-user audits are not designed to be as in-depth as formal audits, they allow for continuous, iterative accountability loops between users and the system: users may be able to conduct end-user audits repeatedly over time as the system evolves.

6.2 Limitations & Future work

6.2.1 Practical implementation concerns. We first acknowledge several concerns related to the practical deployment of our methodology. First, our approach assumes that users are highly motivated to devote time and effort to conduct their own audit. While prior work suggests that users already engage in such behavior [66], as with other issue reporting mechanisms, our method is subject to self-selection bias. Even though any user could use our tool in theory, more privileged or tech-savvy users may be over-represented in practice. Developers who adopt our approach must be aware of this bias and take active steps to ensure that marginalized users are represented.

Second, there are questions about whether our approach depends on buy-in from companies. Would companies be willing to provide the data for an end-user audit? Would companies want to expose themselves to the risk of users uncovering problematic behavior? Our approach is possible without company buy-in (a third-party individual could scrape a platform to collect data and gather annotations using a crowdsourcing platform). However, it is advantageous if companies cooperate with end-user audits because (1) they can provide the data that they already hold, (2) they can host

the end-user auditing site and integrate it directly into their platform, and (3) they can serve as the recipients of audit reports and form a direct line of communication to address issues raised by end-user auditors (rather than requiring reports to first gain attention in the public eye).

While we cannot guarantee that companies would adopt end-user auditing, work on second-party audits [72] indicates that companies want to audit their systems and value the expertise of outside parties. Companies want to know how their algorithmic systems are working, but they don't want *others* to know if they are not working well. Initiatives like bug bounty and bias bounty programs [15, 22] demonstrate mechanisms for end users to surface issues while only alerting system developers to these issues. In line with these programs, companies can integrate end-user auditing into their workflow as a standard pre-launch testing procedure. As soon as they have a model and dataset, they can recruit a set of stakeholders and members of marginalized groups to catch unanticipated, harmful system errors before launch. There is much opportunity for future work not only on the technical and design side, but on the legal and regulatory side to compel companies to implement such auditing capabilities into their systems.

Third, while a developer can inspect a moderate number of end-user audits, once the number of audits scales up to the hundreds or thousands, it becomes much more challenging to manually review these audits. An important area of future work will be to effectively and fairly aggregate the audit reports that users generate. Finally, it is important to acknowledge that end-user audits, just like algorithm audits in general, are not a cure-all. Audits cannot fix inherently problematic or unethical systems that should not exist at all. It is critical that developers who leverage end-user audits interrogate the broader ethical risks of their systems.

6.2.2 Technical failings may limit users. Next, our end-user model is imperfect and cannot be expected to fully capture the complexities of user perspectives for difficult tasks like content moderation. We aim for our model to serve as a helpful guide, but it is possible that this model itself may be biased (for example, if the initial set of annotators is biased or fails to represent a new user's views). While our system may be inaccurate in estimating an end user's labels, our approach is at least a *conservative* estimate of the extent of bias. If the user is not similar to the initial annotators, their recommender system model will borrow information from annotators who are closer to the majority view, which will result in a more majoritarian model that yields a more conservative estimate of system error. A benefit of our collaborative filtering approach (not using content features) is that if the target system makes errors based on features of the content, our model will not make the same errors. Collaborative filtering only draws upon inter-person labeling patterns and thus results in less correlation between the errors of our model and those of the target system. Developers who adopt end-user auditing must take care that their datasets are annotated by a diverse set of users who are representative of the user population (which is of course crucial in developing their system in the first place). Future work might explore more advanced recommender system models that better capture user opinions.

6.2.3 Audit rigor. While our auditing approach shows promise in enabling non-technical users to effectively lead audits, there are opportunities to improve the rigor of these audits. First, our modeling approach is limited to the set of examples in a collected dataset, so we cannot predict user's ratings of unseen examples. Future work might expand this method to project user's opinions on novel examples. Another potential problem is that users might raise spurious issues if they misinterpret plots or raise audit reports based on limited evidence. Prior exploratory research has identified this difficulty in discerning the legitimacy of reports as a core challenge for user-led auditing [19]. In the worst case, our approach defaults back to the mode of bug reporting where users may raise one-off issues. A benefit of our approach is that users *can* collect a large body of

evidence and explain their reasoning, and developers can easily "see through the user's eyes" and look at the same plots and predictions that the user saw to assess the validity of their report.

7 CONCLUSION

Algorithm audits are powerful tools for accountability, but they are currently wielded by a small group of technical experts. While these auditors can successfully advocate on behalf of users, they cannot know the lived experience of all users to catch all potential algorithmic harms they face. If end users could lead their own audits, they could speak for themselves and catch problematic system behavior that technical experts might miss. In our work, we introduce *end-user auditing*, an approach that brings into reality this vision of system-scale audits led by individual, non-technical users. Leveraging a collaborative filtering method that builds off of data that most systems already hold, we lower the barrier to algorithm auditing by minimizing the labor of labeling system outputs and providing an interface that orients the auditing process around the user's perspectives. Our evaluation finds that end-user auditors can successfully audit real-world systems and uncover insights that go beyond what has been previously documented by formal audits. End-user audits work toward a mode of algorithmic accountability where individual end users—especially those who are often marginalized or silenced—can speak truth to algorithmic power.

ACKNOWLEDGMENTS

We thank our anonymous reviewers as well as Joseph Seering, Joon Sung Park, Ranjay Krishna, Matthew Jörke, and Helena Vasconcelos for their insightful feedback and suggestions on our paper. We thank Deepak Kumar for providing our toxic comments dataset. Michelle S. Lam was supported by the Brown Institute for Media Innovation at the Stanford School of Engineering.

REFERENCES

[1] Saleema Amershi, Max Chickering, Steven M. Drucker, Bongshin Lee, Patrice Simard, and Jina Suh. 2015. ModelTracker: Redesigning Performance Analysis Tools for Machine Learning. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 337–346. https://doi.org/10.1145/2702123.2702509

[2] Joshua Asplund, Motahhare Eslami, Hari Sundaram, Christian Sandvig, and Karrie Karahalios. 2020. Auditing race and gender discrimination in online housing markets. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 14. 24–35.

[3] Joshua Attenberg, Panos Ipeirotis, and Foster Provost. 2015. Beat the Machine: Challenging Humans to Find a Predictive Model's "Unknown Unknowns". *J. Data and Information Quality* 6, 1, Article 1 (mar 2015), 17 pages. https://doi.org/10.1145/2700832

[4] Jack Bandy. 2021. Problematic Machine Behavior: A Systematic Literature Review of Algorithm Audits. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW1, Article 74 (apr 2021), 34 pages. https://doi.org/10.1145/3449148

[5] Jack Bandy and Nicholas Diakopoulos. 2020. Auditing News Curation Systems: A Case Study Examining Algorithmic and Editorial Logic in Apple News. *Proceedings of the International AAAI Conference on Web and Social Media* 14, 1 (May 2020), 36–47. https://ojs.aaai.org/index.php/ICWSM/article/view/7277

[6] Paul M Barrett. 2020. Who Moderates the Social Media Giants? A Call to End Outsourcing. *NYU Stern Center for Business and Human Rights* (2020).

[7] Ruha Benjamin. 2019. *Race After Technology: Abolitionist Tools for the New Jim Code.* Polity.

[8] Shea Brown, Jovana Davidovic, and Ali Hasan. 2021. The algorithm audit: Scoring the algorithms that score us. *Big Data & Society* 8, 1 (2021), 2053951720983865. https://doi.org/10.1177/2053951720983865 arXiv:https://doi.org/10.1177/2053951720983865

[9] Joy Buolamwini and Timnit Gebru. 2018. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency (Proceedings of Machine Learning Research, Vol. 81)*, Sorelle A. Friedler and Christo Wilson (Eds.). PMLR, 77–91. https://proceedings.mlr.press/v81/buolamwini18a.html

[10] Ángel Alexander Cabrera, Abraham J. Druck, Jason I. Hong, and Adam Perer. 2021. Discovering and Validating AI Errors With Crowdsourced Failure Reports. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW2, Article 425 (Oct 2021),

22 pages. https://doi.org/10.1145/3479569

[11] Robyn Caplan and Tarleton Gillespie. 2020. Tiered Governance and Demonetization: The Shifting Terms of Labor and Compensation in the Platform Economy. *Social Media + Society* 6, 2 (2020), 2056305120936636. https://doi.org/10.1177/2056305120936636 arXiv:https://doi.org/10.1177/2056305120936636

[12] Kathy Charmaz. 2006. *Constructing grounded theory: A practical guide through qualitative analysis.* Sage.

[13] 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.* ACM Press, Montreal QC, Canada, 1–14. https://doi.org/10.1145/3173574.3174225

[14] Le Chen, Alan Mislove, and Christo Wilson. 2015. Peeking beneath the hood of uber. In *Proceedings of the 2015 internet measurement conference.* 495–508.

[15] Rumman Chowdhury and Jutta Williams. 2021. Introducing Twitter's first algorithmic bias bounty challenge. *Twitter Engineering Blog* (2021). https://blog.twitter.com/engineering/en_us/topics/insights/2021/algorithmic-bias-bounty-challenge

[16] Sasha Costanza-Chock, Inioluwa Deborah Raji, and Joy Buolamwini. 2022. Who Audits the Auditors? Recommendations from a Field Scan of the Algorithmic Auditing Ecosystem. In *2022 ACM Conference on Fairness, Accountability, and Transparency* (Seoul, Republic of Korea) *(FAccT '22).* Association for Computing Machinery, New York, NY, USA, 1571–1583. https://doi.org/10.1145/3531146.3533213

[17] Cédric Courtois, Laura Slechten, and Lennert Coenen. 2018. Challenging Google Search filter bubbles in social and political information: Disconforming evidence from a digital methods case study. *Telematics and Informatics* 35, 7 (2018), 2006–2015. https://doi.org/10.1016/j.tele.2018.07.004

[18] Kimberlé Crenshaw. 1989. Demarginalizing the Intersection of Race and Sex: A Black Feminist Critique of Antidiscrimination Doctrine, Feminist Theory and Antiracist Politics. *University of Chicago Legal Forum* (1989), 139–168.

[19] Alicia DeVos, Aditi Dhabalia, Hong Shen, Kenneth Holstein, and Motahhare Eslami. 2022. Toward User-Driven Algorithm Auditing: Investigating users' strategies for uncovering harmful algorithmic behavior. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22).* Association for Computing Machinery, New York, NY, USA, 19 pages. https://doi.org/10.1145/3491102.3517441

[20] Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and Mitigating Unintended Bias in Text Classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society* (New Orleans, LA, USA) *(AIES '18).* Association for Computing Machinery, New York, NY, USA, 67–73. https://doi.org/10.1145/3278721.3278729

[21] Benjamin Edelman, Michael Luca, and Dan Svirsky. 2017. Racial discrimination in the sharing economy: Evidence from a field experiment. *American Economic Journal: Applied Economics* 9, 2 (2017), 1–22.

[22] Ira Globus-Harris, Michael Kearns, and Aaron Roth. 2022. An Algorithmic Framework for Bias Bounties. In *2022 ACM Conference on Fairness, Accountability, and Transparency* (Seoul, Republic of Korea) *(FAccT '22).* Association for Computing Machinery, New York, NY, USA, 1106–1124. https://doi.org/10.1145/3531146.3533172

[23] Mitchell L. Gordon, Michelle S. Lam, Joon Sung Park, Kayur Patel, Jeff Hancock, Tatsunori Hashimoto, and Michael S. Bernstein. 2022. Jury Learning: Integrating Dissenting Voices into Machine Learning Models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22).* Association for Computing Machinery, New York, NY, USA, Article 115, 19 pages. https://doi.org/10.1145/3491102.3502004

[24] Mitchell L. Gordon, Kaitlyn Zhou, Kayur Patel, Tatsunori Hashimoto, and Michael S. Bernstein. 2021. *The Disagreement Deconvolution: Bringing Machine Learning Performance Metrics In Line With Reality.* Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3411764.3445423

[25] Robert Gorwa, Reuben Binns, and Christian Katzenbach. 2020. Algorithmic content moderation: Technical and political challenges in the automation of platform governance. *Big Data & Society* 7, 1 (2020), 2053951719897945. https://doi.org/10.1177/2053951719897945 arXiv:https://doi.org/10.1177/2053951719897945

[26] Tommi Gröndahl, Luca Pajola, Mika Juuti, Mauro Conti, and N. Asokan. 2018. All You Need is "Love": Evading Hate Speech Detection. In *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security* (Toronto, Canada) *(AISec '18).* Association for Computing Machinery, New York, NY, USA, 2–12. https://doi.org/10.1145/3270101.3270103

[27] Maarten Grootendorst. 2020. BERTopic: Leveraging BERT and c-TF-IDF to create easily interpretable topics. https://doi.org/10.5281/zenodo.4381785

[28] Oliver L. Haimson, Daniel Delmonaco, Peipei Nie, and Andrea Wegner. 2021. Disproportionate Removals and Differing Content Moderation Experiences for Conservative, Transgender, and Black Social Media Users: Marginalization and Moderation Gray Areas. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW2, Article 466 (Oct 2021), 35 pages. https://doi.org/10.1145/3479610

[29] Alex Hanna, Emily Denton, Andrew Smart, and Jamila Smith-Loud. 2020. Towards a Critical Race Methodology in Algorithmic Fairness. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (Barcelona, Spain) *(FAT* '20).* Association for Computing Machinery, New York, NY, USA, 501–512. https://doi.org/10.1145/

3351095.3372826

[30] Aniko Hannak, Gary Soeller, David Lazer, Alan Mislove, and Christo Wilson. 2014. Measuring Price Discrimination and Steering on E-Commerce Web Sites. In *Proceedings of the 2014 Conference on Internet Measurement Conference* (Vancouver, BC, Canada) *(IMC '14)*. Association for Computing Machinery, New York, NY, USA, 305–318. https://doi.org/10.1145/2663716.2663744

[31] Camille Harris, Matan Halevy, Ayanna Howard, Amy Bruckman, and Diyi Yang. 2022. Exploring the Role of Grammar and Word Choice in Bias Toward African American English (AAE) in Hate Speech Classification. In *2022 ACM Conference on Fairness, Accountability, and Transparency* (Seoul, Republic of Korea) *(FAccT '22)*. Association for Computing Machinery, New York, NY, USA, 789–798. https://doi.org/10.1145/3531146.3533144

[32] Anna Lauren Hoffmann. 2019. Where fairness fails: data, algorithms, and the limits of antidiscrimination discourse. *Information, Communication & Society* 22, 7 (2019), 900–915. https://doi.org/10.1080/1369118X.2019.1573912

[33] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé, Miro Dudik, and Hanna Wallach. 2019. *Improving Fairness in Machine Learning Systems: What Do Industry Practitioners Need?* Association for Computing Machinery, New York, NY, USA, 1–16. https://doi.org/10.1145/3290605.3300830

[34] Hossein Hosseini, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. 2017. Deceiving Google's Perspective API Built for Detecting Toxic Comments. arXiv:1702.08138 [cs.LG]

[35] Jiachen Jiang and Soroush Vosoughi. 2020. Not Judging a User by Their Cover: Understanding Harm in Multi-Modal Processing within Social Media Research. In *Proceedings of the 2nd International Workshop on Fairness, Accountability, Transparency and Ethics in Multimedia* (Seattle, WA, USA) *(FATE/MM '20)*. Association for Computing Machinery, New York, NY, USA, 6–12. https://doi.org/10.1145/3422841.3423534

[36] Jialun Aaron Jiang, Morgan Klaus Scheuerman, Casey Fiesler, and Jed R. Brubaker. 2021. Understanding international perceptions of the severity of harmful content online. *PLOS ONE* 16, 8 (08 2021), 1–22. https://doi.org/10.1371/journal.pone.0256762

[37] Jigsaw. 2019. Jigsaw Unintended Bias in Toxicity Classification. https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/overview

[38] Jigsaw. 2022. *Perspective API.* Retrieved January 2, 2022 from https://www.perspectiveapi.com/

[39] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2018. Preventing Fairness Gerrymandering: Auditing and Learning for Subgroup Fairness. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, 2564–2572. https://proceedings.mlr.press/v80/kearns18a.html

[40] Allison Koenecke, Andrew Nam, Emily Lake, Joe Nudell, Minnie Quartey, Zion Mengesha, Connor Toups, John R. Rickford, Dan Jurafsky, and Sharad Goel. 2020. Racial disparities in automated speech recognition. *Proceedings of the National Academy of Sciences* 117, 14 (2020), 7684–7689. https://doi.org/10.1073/pnas.1915768117 arXiv:https://www.pnas.org/content/117/14/7684.full.pdf

[41] Deepak Kumar, Patrick Gage Kelley, Sunny Consolvo, Joshua Mason, Elie Bursztein, Zakir Durumeric, Kurt Thomas, and Michael Bailey. 2021. Designing Toxic Content Classification for a Diversity of Perspectives. In *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*. USENIX Association, 299–318. https://www.usenix.org/conference/soups2021/presentation/kumar

[42] Cameron Lai and Markus Luczak-Roesch. 2019. You Can't See What You Can't See: Experimental Evidence for How Much Relevant Information May Be Missed Due to Google's Web Search Personalisation. In *Social Informatics*, Ingmar Weber, Kareem M. Darwish, Claudia Wagner, Emilio Zagheni, Laura Nelson, Samin Aref, and Fabian Flöck (Eds.). Springer International Publishing, Cham, 253–266.

[43] Michael Madaio, Lisa Egede, Hariharan Subramonyam, Jennifer Wortman Vaughan, and Hanna Wallach. 2021. Assessing the Fairness of AI Systems: AI Practitioners' Processes, Challenges, and Needs for Support. arXiv:2112.05675 [cs.AI]

[44] Brandeis Marshall. 2021. *Algorithmic Misogynoir in Content Moderation Practice.* Technical Report. Technical Report. Heinrich-Böll-Stiftung.

[45] J. Nathan Matias, Austin Hounsel, and Nick Feamster. 2021. Software-Supported Audits of Decision-Making Systems: Testing Google and Facebook's Political Advertising Policies. arXiv:2103.00064 [cs.HC]

[46] Louise Matsakis and Paris Martineau. 2020. Coronavirus disrupts social media's first line of defense. *WIRED Magazine* 18 (2020).

[47] Jesse McCrosky and Brandi Geurkink. 2021. *YouTube Regrets: A crowdsourced investigation into YouTube's recommendation algorithm.* Retrieved January 2, 2022 from https://foundation.mozilla.org/en/youtube/findings/

[48] Nora McDonald, Sarita Schoenebeck, and Andrea Forte. 2019. Reliability and Inter-Rater Reliability in Qualitative Research: Norms and Guidelines for CSCW and HCI Practice. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 72 (nov 2019), 23 pages. https://doi.org/10.1145/3359174

[49] Danaë Metaxa, Joon Sung Park, James A. Landay, and Jeff Hancock. 2019. Search Media and Elections: A Longitudinal Investigation of Political Search Results. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 129 (Nov 2019), 17 pages.

https://doi.org/10.1145/3359231

[50] Danaë Metaxa, Joon Sung Park, Ronald E. Robertson, Karrie Karahalios, Christo Wilson, Jeff Hancock, and Christian Sandvig. 2021. Auditing Algorithms: Understanding Algorithmic Systems from the Outside In. *Foundations and Trends® in Human–Computer Interaction* 14, 4 (2021), 272–344. https://doi.org/10.1561/1100000083

[51] Jakub Mikians, László Gyarmati, Vijay Erramilli, and Nikolaos Laoutaris. 2013. Crowd-Assisted Search for Price Discrimination in e-Commerce: First Results. In *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies* (Santa Barbara, California, USA) *(CoNEXT '13)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/2535372.2535415

[52] Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. 2021. Dealing with Disagreements: Looking Beyond the Majority Vote in Subjective Annotations. *arXiv e-prints*, Article arXiv:2110.05719 (Oct. 2021), arXiv:2110.05719 pages. arXiv:2110.05719 [cs.CL]

[53] Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English Tweets. arXiv:2005.10200 [cs.CL]

[54] Safiya Umoja Noble. 2018. *Algorithms of oppression.* New York University Press.

[55] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. 2019. Dissecting racial bias in an algorithm used to manage the health of populations. *Science* 366, 6464 (2019), 447–453. https://doi.org/10.1126/science.aax2342 arXiv:https://www.science.org/doi/pdf/10.1126/science.aax2342

[56] Ihudiya Finda Ogbonnaya-Ogburu, Angela D.R. Smith, Alexandra To, and Kentaro Toyama. 2020. *Critical Race Theory for HCI.* Association for Computing Machinery, New York, NY, USA, 1–16. https://doi.org/10.1145/3313831.3376392

[57] Vinodkumar Prabhakaran, Aida Mostafazadeh Davani, and Mark Díaz. 2021. On Releasing Annotator-Level Labels and Information in Datasets. arXiv:2110.05699 [cs.CL]

[58] Inioluwa Deborah Raji and Joy Buolamwini. 2019. Actionable Auditing: Investigating the Impact of Publicly Naming Biased Performance Results of Commercial AI Products. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society* (Honolulu, HI, USA) *(AIES '19)*. Association for Computing Machinery, New York, NY, USA, 429–435. https://doi.org/10.1145/3306618.3314244

[59] Inioluwa Deborah Raji, Andrew Smart, Rebecca N. White, Margaret Mitchell, Timnit Gebru, Ben Hutchinson, Jamila Smith-Loud, Daniel Theron, and Parker Barnes. 2020. Closing the AI Accountability Gap: Defining an End-to-End Framework for Internal Algorithmic Auditing. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (Barcelona, Spain) *(FAT* '20)*. Association for Computing Machinery, New York, NY, USA, 33–44. https://doi.org/10.1145/3351095.3372873

[60] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://arxiv.org/abs/1908.10084

[61] Ronald E. Robertson, Shan Jiang, Kenneth Joseph, Lisa Friedland, David Lazer, and Christo Wilson. 2018. Auditing Partisan Audience Bias within Google Search. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 148 (nov 2018), 22 pages. https://doi.org/10.1145/3274417

[62] Russ R Salakhutdinov and Andriy Mnih. 2008. Probabilistic Matrix Factorization. In *Advances in Neural Information Processing Systems*, J. Platt, D. Koller, Y. Singer, and S. Roweis (Eds.), Vol. 20. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2007/file/d7322ed717dedf1eb4e6e52a37ea7bcd-Paper.pdf

[63] Christian Sandvig, Kevin Hamilton, Karrie Karahalios, and Cedric Langbort. 2014. Auditing Algorithms: Research Methods for Detecting Discrimination on Internet Platforms. *Data and Discrimination: Converting Critical Concerns into Productive Inquiry* (2014), 1–23.

[64] Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A Smith. 2019. The Risk of Racial Bias in Hate Speech Detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 1668–1678.

[65] Ari Schlesinger, W. Keith Edwards, and Rebecca E. Grinter. 2017. Intersectional HCI: Engaging Identity through Gender, Race, and Class. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 5412–5427. https://doi.org/10.1145/3025453.3025766

[66] Hong Shen, Alicia DeVos, Motahhare Eslami, and Kenneth Holstein. 2021. Everyday Algorithm Auditing: Understanding the Power of Everyday Users in Surfacing Harmful Algorithmic Behaviors. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW2, Article 433 (oct 2021), 29 pages. https://doi.org/10.1145/3479577

[67] Márcio Silva, Lucas Santos de Oliveira, Athanasios Andreou, Pedro Olmo Vaz de Melo, Oana Goga, and Fabricio Benevenuto. 2020. *Facebook Ads Monitor: An Independent Auditing System for Political Ads on Facebook.* Association for Computing Machinery, New York, NY, USA, 224–234. https://doi.org/10.1145/3366423.3380109

[68] C. Estelle Smith, Bowen Yu, Anjali Srivastava, Aaron Halfaker, Loren Terveen, and Haiyi Zhu. 2020. Keeping Community in the Loop: Understanding Wikipedia Stakeholder Values for Machine Learning-Based Systems. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing

Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376783
[69] Jesper E Van Engelen and Holger H Hoos. 2020. A survey on semi-supervised learning. *Machine Learning* 109 (2020), 373–440. https://doi.org/10.1007/s10994-019-05855-6
[70] Emily A Vogels. 2021. The State of Online Harassment. *Pew Research Center* 13 (2021).
[71] Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. 2021. DCN V2: Improved Deep & Cross Network and Practical Lessons for Web-scale Learning to Rank Systems. In *Proceedings of the Web Conference 2021*. 1785–1797.
[72] Christo Wilson, Avijit Ghosh, Shan Jiang, Alan Mislove, Lewis Baker, Janelle Szary, Kelly Trindel, and Frida Polli. 2021. Building and Auditing Fair Algorithms: A Case Study in Candidate Screening. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (Virtual Event, Canada) *(FAccT '21)*. Association for Computing Machinery, New York, NY, USA, 666–677. https://doi.org/10.1145/3442188.3445928
[73] Austin P. Wright, Omar Shaikh, Haekyu Park, Will Epperson, Muhammed Ahmed, Stephane Pinel, Duen Horng (Polo) Chau, and Diyi Yang. 2021. RECAST: Enabling User Recourse and Interpretability of Toxicity Detection Models with Interactive Visualization. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW1, Article 181 (apr 2021), 26 pages. https://doi.org/10.1145/3449280
[74] Meg Young, Lassana Magassa, and Batya Friedman. 2019. Toward inclusive tech policy design: a method for underrepresented voices to strengthen tech policy documents. *Ethics and Information Technology* 21 (2019), 89–103.

A FREE-FORM AUDIT PROMPTING QUESTIONS

For the free-form audit task in the user evaluation, study participants were shown the following set of prompting questions and could optionally use these as starting points for their audits.

- Are there terms that are used in your identity group or community that tend to be flagged incorrectly as toxic?
- Are there terms that are used in your identity group or community that tend to be flagged incorrectly as non-toxic?
- Are there certain ways that your community tends to be targeted by outsiders?
- Are there other communities whose content should be very similar to your community's? Verify that this content is treated similarly by the system.
- Are there ways that you've seen individuals in your community actively try to thwart the rules of automated content moderation systems? Check whether these strategies work here.

B FRAMING SCENARIO

To preface the study, we provided participants with the following framing scenario. They would be inspecting a content moderation system for YouSocial, a new social media platform. Under the hood, this platform uses a model that's actually used on real-world sites like The New York Times and The Financial Times. The developer of this content moderation system has built *IndieLabel*, an auditing tool to enable platform users to investigate the behavior of the system and raise potential issues they see. As users of this tool, the errors they discover and share via reports will be directly used by developers to improve the system. We then provided users with an introduction to the 0–4 toxicity rating scheme and explained that on YouSocial, comments with a score lower than 2 ("Moderately toxic") would be deemed non-toxic and would be allowed to remain on the platform while comments with a score of 2 or above would be deemed toxic and would be deleted from the platform. We tied the rating scheme to concrete platform actions to help users provide ratings that were grounded in real-world implications that they could easily reason about.

C PRE- AND POST-AUDIT SURVEY QUESTIONS

C.1 Sign-up survey

C.1.1 Demographics.

(1) Age range (single-select options: 18 to 24, 25 to 34, 35 to 44, 45 to 54, 55 to 64, 65 and above)

(2) Gender (select all that apply: Woman, Man, Non-binary, Prefer to self-describe: _____)
(3) Race (select all that apply: White, Black or African-American, American Indian or Alaska Native, Asian or Asian-American, Hispanic, Native Hawaiian or Pacific Islander, Prefer to self-describe: _____)
(4) Political leaning (single-select options: Very liberal, Liberal, Slightly liberal, Moderate, Slightly conservative, Conservative, Very conservative, None of these describe my political leanings)
(5) Political affiliation (single-select options: Conservative, Independent, Liberal, Other: _____)
(6) Do you identify with the LGBTQ+ community? (Yes/No)
(7) How important is religion to you? (single-select options: Not important, Not too important, Somewhat important, Very important)

C.1.2 Technical background.

(1) How much do you understand machine learning?
(2) How much do you understand algorithms?

These questions had the following 5 options: (a) Not at all (have mostly just heard about the term), (b) Only a little (have read some non-technical materials or explainers on the topic), (c) To some extent (have read technical materials or explainers on the topic), (d) Rather much (have taken a course on this topic), (e) Very much (have taken multiple courses on this topic)

C.2 Pre-audit survey

C.2.1 Perceptions of content moderation systems.

(1) Please select all social media platforms that you use at least once a week (Options: Twitter, Instagram, Reddit, Facebook, TikTok, YouTube, Snapchat, Other: _____)
(2) Have you ever had your content taken down (even if incorrectly) by an automated content moderation system? (Yes/No)
(3) How well do you think existing content moderation systems work for you? (7-point Likert)
(4) What, if anything, would you change about content moderation systems to make them work better for you? (text response)
(5) Based on your experience, toxic posts or comments are _____ a problem (5-point frequency Likert)
(6) What does the term "toxic comment" mean to you? (text response)
(7) How confident are you that you could perform an audit of a content moderation system? (7-point Likert)

C.3 Post-audit survey

C.3.1 Views on the content moderation system.

(1) How well do you think this content moderation system works for you? (7-point Likert)
(2) How well do you think this content moderation system works for the general population? (7-point Likert)
(3) How well do you think you understand the behavior of the content moderation system after this audit? (7-point Likert)
(4) To what extent do you feel that your voice is adequately represented by the content moderation system? (7-point Likert)
(5) If you knew that the YouSocial platform would be using your audit report to make changes, to what extent would you trust its content moderation system? (7-point Likert)

C.3.2 Views on the auditing tool.

(1) How comfortable were you with auditing the content moderation system? (7-point Likert)

(2) In using this auditing tool, how would you rate your level of control to guide the audit? (7-point Likert)
(3) In using this auditing tool, how would you rate your ability to express your own voice through your report? (7-point Likert)
(4) Did you feel that you discovered potential issues with the system you hadn't anticipated? (Yes/No)
(5) How likely do you think you would have discovered these same issues without a tool like the one you used today? (ex: through manual inspection of system outputs or regular usage of a site) (7-point Likert)
(6) How useful is this tool in aiding understanding of the *overall, system-wide behavior* of a content moderation system? (7-point Likert)
(7) How useful is this tool in enabling you to *capture your personal values* (via your personalized model)? (7-point Likert)
(8) What did you like about your auditing experience? Please describe the positives of this tool. (text response)
(9) What did you not like about your auditing experience? Please describe the negatives of this tool or any issues you experienced. (text response)

C.3.3 Views on the end-user versus group-based model.

(1) How difficult did you find the task of training your personal model? (7-point Likert)
(2) To what extent do you feel that the *personalized model* helped you to focus your attention on potential issues that you find important? (7-point Likert)
(3) To what extent do you feel that the *group-based baseline model* helped you to focus your attention on potential issues that you find important? (7-point Likert)
(4) Based on your experience, which of these options would you prefer to audit a content moderation system? (Options: Only a personalized model, Only a group-based model, Both a personalized model and group-based model, Neither a personalized model nor a group-based model)
(5) How would you compare your auditing experience using your personalized model versus using the group-based baseline model? Did one better capture your views than the other? (text response)
(6) For the sites you use, have you ever seen comments similar to the ones we showed you? (Yes/No)
(7) Have you ever personally been the target of comments similar to the ones you reviewed? (Yes/No)

D ADDITIONAL END-USER AUDIT RESULTS

D.1 User-proposed solutions

In response to harms that they uncovered in their end-user audits, users proposed their own solutions based on their own perspectives and experience, as summarized in Table 5.

D.2 Audit topics investigated by multiple participants

Here in Figure 9, we summarize the set of topics that were audited by more than one study participant.

D.3 User perceptions of *IndieLabel*'s helpfulness

In Figure 10, we summarize participant survey responses related to the helpfulness of the *IndieLabel* tool, separating their impressions of the end-user model and group-based model.

Suggested solution type	Explanation
Consider the target of speech	Comments can take on very different meanings depending on whether they are targeted at another user (and may further vary based on the relationship between the users), a group, oneself, or a non-person (like a place, object, or corporation).
Consider the identity of the poster	The identity of the poster also plays a large role in shaping the meaning of a comment. If the comment is directed at a group, the poster's membership or outsider status to the group may determine whether the comment is harmful.
Disambiguate multiple senses of keywords	Many keywords often associated with toxic content have multiple differing meanings. More sophisticated disambiguation between these different usages would likely reduce over-flagging behavior.
Differentiate between triggering content and toxic content	In discussions about sensitive topics, comments may be triggering for users. However, participants felt that these kinds of content should not be treated the same way as toxic content because erasure of these discussions would be harmful to marginalized groups. There are other options besides content removal, and alternative designs may help to foster wellbeing while keeping discussions alive.
Consider the tone, mood, or emotion of the comment	The emotion or tone of a comment often significantly shapes its interpretation. Without a grasp on the emotions underlying a comment, content moderation systems may overflag humorous or emotive wordings.

Table 5. Potential solutions raised by study participants (summarized into main categories)

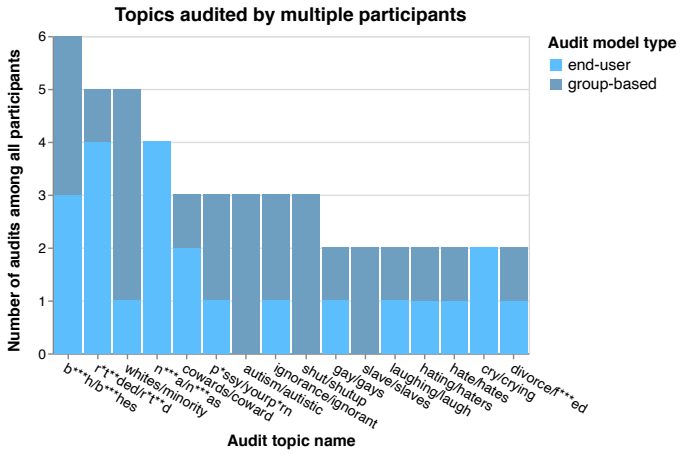

Fig. 9. Summary of the 16 topics investigated by *multiple* study participants for the fixed audits.

E DEEP LEARNING MODEL

We use TensorFlow Recommenders (TFRS) as the basis of our implementation. TFRS natively supports Deep & Cross Networks (DCNs) [71]. We use Huggingface's Tensorflow API to instantiate BERTweet (a large-scale language model pre-trained on English Tweets, released by NVIDIA [53]) as pre-trained content embeddings within our recommender system. We adapt the model to the task by performing an initial fine-tuning step on a large-scale toxicity dataset released by Jigsaw [37]. We co-train the entire model for two epochs, freeze the large language model, and continue training the remainder of the model for 8 epochs. Further epochs did not noticeably improve the model's performance. We used the Adam optimizer and Mean Squared Error as our loss function.

We trained our model on one machine with one NVIDIA Titan XP GPU. We chose standard hyperparameters used when fine tuning BERT-based models: we used learning rate of $2e-5$, a batch size of 16, and a maximum length of 128 tokens. We set our DCN-specific hyperparameters as follows: we set a constant embedding dimension of 32, a three-layer cross network of size 768,

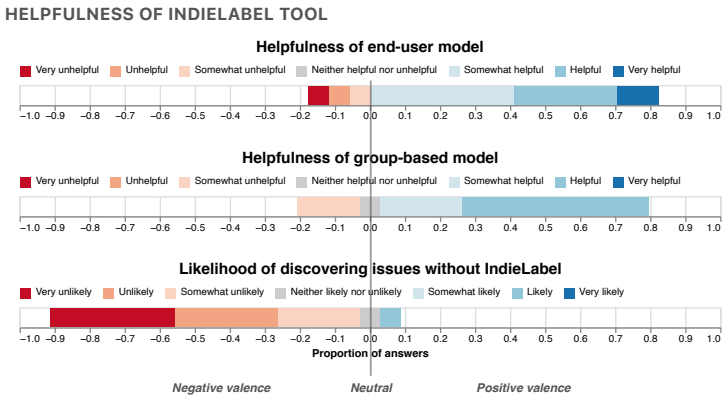

Fig. 10. Perceptions of the helpfulness of *IndieLabel*. Users found both the end-user and group-based model helpful, and only a small minority of users felt they might have discovered the same issues without the tool.

three dense layers of size 768, and a output dense layer of size 1. We selected these sizes and the number of training epochs after performing a small grid search.

F TECHNICAL EVALUATION: ADDITIONAL PERFORMANCE METRICS

Following the same methodology used to analyze the performance of the five model variants for the MAE metric, we analyzed the performance for MSE, RMSE, and classification accuracy and observed the same results, as shown in Figures 11, 12, and 13. Using the same linear mixed-effects model definition for each of these metrics, we observed a significant main effect of model type (MSE: $F(4, 266.12) = 78.39, p < .001$; RMSE: $F(4, 266.1) = 108.34, p < .001$; Accuracy: $F(4, 266.31) = 51.83, p < .001$). Our posthoc pairwise Tukey test found statistically significant ($p < .05$) differences between the mean metric values of all pairs of models except for SVD uniform vs. end-user for MSE and SVD uniform vs. group for accuracy. We again note that there was a significant performance improvement of both end-user model variants over their corresponding group-based models ($p < .01$ for both the SVD and DL variants).

Received January 2022; revised April 2022; accepted August 2022

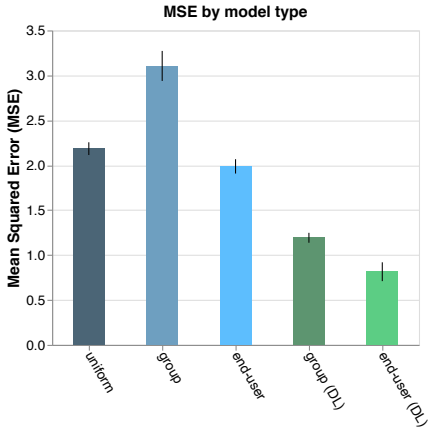

Fig. 11. Mean squared error on participants' test sets (on 0–4 score range). The two end-user model variants achieve lower MSE compared to their corresponding group-based models.

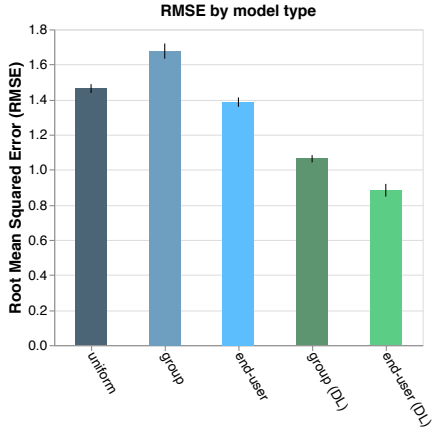

Fig. 12. Root mean squared error on participants' test sets (on 0–4 score range). The two end-user model variants achieve lower RMSE compared to their corresponding group-based models.

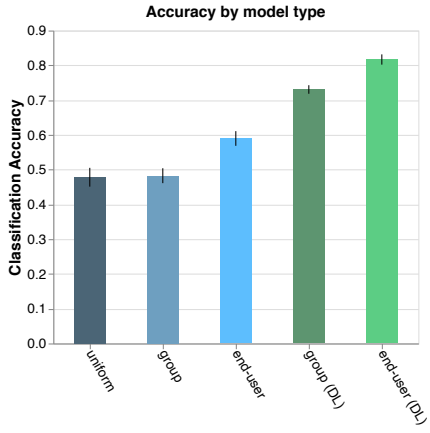

Fig. 13. Classification accuracy on participants' test sets (on 0–4 score range). The two end-user model variants achieve higher accuracy compared to their corresponding group-based models.