Wikipedia ORES Explorer: Visualizing Trade-offs For Designing Applications With Machine Learning API

Zining Ethan Ye
Human-Computer Interaction Institute
Carnegie Mellon University
Pittsburgh, PA, United States
ethanye0512@gmail.com

Xinran Yuan
Human-Computer Interaction Institute
Carnegie Mellon University
Pittsburgh, PA, United States
yuanxinran@live.com

Shaurya Gaur
School of Electrical Engineering and Computer Science
Oregon State University
Corvallis, Oregon, United States
shauryavrat@live.com

Aaron Halfaker
Microsoft
Redmond, WA, USA
ahalfaker@microsoft.com

Jodi Forlizzi
Human-Computer Interaction Institute
Carnegie Mellon University
Pittsburgh, PA, United States
achould@cmu.edu

Haiyi Zhu
Human-Computer Interaction Institute
Carnegie Mellon University
Pittsburgh, PA, United States
haiyiz@cs.cmu.edu

ABSTRACT

With the growing industry applications of Artificial Intelligence (AI) systems, pre-trained models and APIs have emerged and greatly lowered the barrier of building AI-powered products. However, novice AI application designers often struggle to recognize the inherent algorithmic trade-offs and evaluate model fairness before making informed design decisions. In this study, we examined the Objective Revision Evaluation System (ORES), a machine learning (ML) API in Wikipedia used by the community to build anti-vandalism tools. We designed an interactive visualization system to communicate model threshold trade-offs and fairness in ORES. We evaluated our system by conducting 10 in-depth interviews with potential ORES application designers. We found that our system helped application designers who have limited ML backgrounds learn about in-context ML knowledge, recognize inherent value trade-offs, and make design decisions that aligned with their goals. By demonstrating our system in a real-world domain, this paper presents a novel visualization approach to facilitate greater accessibility and human agency in AI application design.

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**; *Visualization techniques*; **Collaborative and social computing**.

KEYWORDS

Social Computing, Algorithmic Trade-offs, Wikipedia, AI explainability, Interactive visualization

DIS '21, June 28-July 2, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8476-6/21/06…$15.00
https://doi.org/10.1145/3461778.3462099

ACM Reference Format:
Zining Ethan Ye, Xinran Yuan, Shaurya Gaur, Aaron Halfaker, Jodi Forlizzi, and Haiyi Zhu. 2021. Wikipedia ORES Explorer: Visualizing Trade-offs For Designing Applications With Machine Learning API. In *Designing Interactive Systems Conference 2021 (DIS '21), June 28-July 2, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3461778.3462099

1 INTRODUCTION

While developing and training machine learning (ML) models can involve complex processes, the building of machine learning powered applications is now expanding to ML novices. Machine learning capabilities are increasingly offered as industry services and are becoming accessible to people with limited ML expertise. For example, platforms such as Amazon Sagemaker Autopilot[1], Google Cloud AutoML[2], Microsoft Azure[3] and IBM AutoAI[4] provide automatic pipelines through which ML novices can develop machine learning models to address real-world problems. In other cases, machine learning is offered as an API that people can directly call to access pre-trained models and get prediction results.

Recent research [35, 36] investigated how application designers build ML solutions and identified two unique challenges in envisioning and prototyping with AI: the uncertainty surrounding AI's capabilities, and AI's output complexity. Specifically, there are often inherent trade-offs between different system criteria of machine learning models (e.g., accuracy, false-positive rates, false-negative rates, and disparity) and optimizing one criterion often leads to poor performance in others. Furthermore, there is an emerging body of literature demonstrating that machine learning models can have disparate performances, for example, among different social groups [3, 20, 24]. This disparity can impact experiences of users, and even lead to negative societal outcomes [21]. However, there is little research on how to help application designers understand the

[1] Amazon Sagemaker Autopilot: https://aws.amazon.com/sagemaker/autopilot/
[2] Google Cloud AutoML: https://cloud.google.com/automl
[3] Microsoft Azure Machine Learning: https://azure.microsoft.com/en-us/services/machine-learning/
[4] IBM AutoAI: https://www.ibm.com/demos/collection/IBM-Watson-Studio-AutoAI/

trade-offs and disparities in ML services in order to make informed decisions and achieve their design goals.

In this paper, we focus on a service that is designed to help application designers–whom we broadly define as a group of designers, engineers and product managers that are ML novices–to understand the trade-offs in ML models during the design and development process.Wikipedia has a community of application designers commonly referred to as "tool developers" or "bot developers." These application designers are members of the Wikipedia community and are often prolific editors themselves. They develop technologies that other Wikipedians use as *tools* to support the contribution and curation work in Wikipedia. The Objective Revision Evaluation Service (ORES) is a web service and API developed by Wikimedia's Scoring Platform Team. The service allows users to build applications to fight vandalism and review edits on Wikipedia and is available to Wikipedia application designers around the world. It uses machine learning to evaluate the quality and intention of Wikipedia edits [16]. The ORES API takes in an edit revision ID and outputs both a "damaging" and "good-faith" score that represent the respective likelihoods that an edit is damaging or malicious. Using ORES, application designers have the opportunity to decide how to use the prediction scores, choosing, for example, a prediction threshold for their applications. While building effective ORES tools requires deep contextual knowledge and technical expertise, it is uncommon for these application designers have substantial backgrounds in machine learning or data engineering.

Without a thorough understanding of model trade-offs, application designers may apply ML capabilities to systems that could lead to under-performing and potentially serious problems. Moreover, the ORES model tends to be more aggressive to edits made by newcomers and anonymous editors; false positive rates of edits from those two groups are significantly higher than false positive rates of edits from the experienced editors [16, 28]. Such disparities might impact the experiences of novice users and discourage their edit contributions in the Wikipedia Community. Thus, application designers should be made aware of the inherent trade-offs and model fairness in ML models in order to make informed decisions about selecting desired models, or taking action to mitigate potential problems caused by performance disparity.

In the research that follows, we present a case study that explores using interactive visualizations to communicate Wikipedia's ORES API's threshold-associated trade-offs and model fairness across different groups of editors. To surface the issues discussed above and help application designers effectively use ORES, we designed and implemented ORES Explorer, an interactive visualization website that uses a sample data set to allow people to learn about and experiment with ORES models. There are four visualizations in ORES Explorer: About ORES, Threshold Explorer, Group Disparity Visualizer, and Threshold Recommender. The goal of the visualization website is to eventually assist ORES application designers in making sensible product decisions that align with operational needs.

To evaluate the effectiveness of ORES Explorer, we conducted a series of interviews with application designers using a think-aloud protocol. During the interview, participants were given a scenario for building an ORES-based tool (either an automated bot or a human-review tool) and asked to express their opinions as they explored the visualizations. The goal of each interview was to see whether participants could perceive trade-offs and performance disparity in the ORES system and, eventually, make reasonable decisions based on the types of tools that they were developing. We found that 1) our visualization system improved people's understanding of the trade-offs in setting thresholds for the ORES model; 2) participants were able to select model thresholds that aligned with their design goals based on gathered information; and 3) surfacing the trade-offs and bias of the model helped participants develop more trust in the ORES AI system.

This paper contributes findings on how to help designers make sense of model capabilities and limitations in order to build responsible ML applications in the context of Wikipedia. Wikipedia ORES API, however, is not an isolated case. Numerous similar Machine Learning APIs have been created to help designers build ML solutions. For example, Google's Perspective API[5], which takes in any text-based conversation and outputs toxicity scores, has been used by publishers, platforms, and individuals to power a variety of different use cases of content moderation. We believe that our findings and aspects of our visualization tool are readily generalizable to other contexts.

2 RELATED WORK

2.1 Machine Learning as a Design Material

The application of Artificial Intelligence (AI) has flourished in recent years. Technologies such as facial recognition, data visualization, predictive analytics, natural language processing, and deep learning are becoming more integrated into the design of essential products and services, thus creating new challenges for developers and designers of these systems. As AI/ML gains recognition as a type of design material in Human-Computer Interaction (HCI) [32], HCI researchers and designers have investigated the idea of AI as a design material to explore AI's potential in improving interactions with apps and software systems. Research has focused on how design practitioners work with AI, how they envision and propose new uses for AI, and collaboration with data scientists [11, 14, 17, 33, 34]

Other work has focused on the difficulties and bottlenecks designers, often ML novice, experience while working with AI. Yang et al. and Kaur et al. investigated how non-ML-experts build ML solutions using ML services and tools to address their own real-world problems [19, 36]. They identified pitfalls and unique situations that non-ML-experts experience engaging with these services. A related study identified two sources of human-AI interaction design complexity: ML's capability uncertainty, and output complexity, which respectively affect designers' understanding of the ML system, and how they conceptualize the system's behaviors in order to choreograph its interactions. [35]. Other research has been conducted to understand the challenges UX designers face when working with AI [11] [33]. Findings showed that UX designers saw challenges in three main areas: envisioning what ML might be, working with ML as a design material, and ethical concerns on how to purposefully use ML [11]. Additionally, data scientists were found to over-trust and misuse interpretability tools, and few of the participants were able to accurately describe the visualizations output by these tools [19].

[5]https://www.perspectiveapi.com

Some work has been done to facilitate non-experts with the understanding and use of ML. The field has responded with resources, tools, and processes to support ML novices and design practitioners in designing and working with AI. These materials include curricula to improve data science skills [Patrick Hebron. 2016. Machine learning for designers. (2016)., R. King, E.F. Churchill, and C. Tan. 2017. Designing with Data: Improving the User Experience with A/B Testing. O'Reilly Media.], tools and interfaces that allow for exploration of AI as a design material [Rebecca Fiebrink, Dan Trueman, and Perry R Cook. 2009. A Meta-Instrument for Interactive, On-the-Fly Machine Learning. In NIME. 280–285., Philip van Allen. 2018. Prototyping ways of prototyping AI. Interactions 25, 6 (2018), 46–51.], and human-AI interaction guidelines to think through the implications of AI-related design decisions [Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. 2019. Guidelines for human-AI interaction. In Proceedings of the 2019 chi conference on human factors in computing systems. 1–13., Google PAIR. 2019. People + AI Guidebook. pair.withgoogle.com/guidebook]. Yang et al. [36] provided three general principles on how to guide non-experts to easily build robust ML solutions: (1) "grounding its interaction design in non-experts' intuitive approach to ML"; (2) "scaffolding and safeguarding a robust model building process"; (3) "supporting users of various needs and skill sets, both in terms of ML knowledge and programming skills". While we have useful guidelines for non-experts in the space to work with ML models, there is a lack of domain-specific tools. Domain-specific tools are essential in assisting non-experts make algorithmic decisions during their working processes, as they could face common problems such as inherent trade-offs in implementing multiple design goals in the algorithm [37].

2.2 Explaining AI and Machine Learning algorithms

Bringing explainability to ML models is one of the major ways to help non-experts work effectively and responsibly with those complex systems. Prior research in this space can be categorized into two approaches: explaining an algorithm's individual decisions, and explaining an algorithm's performance and outcomes.

2.2.1 Explaining individual decisions. Many studies have been done to explore algorithmic approaches that help humans better understand the results of ML algorithms with the goal of evaluating the model's decisions [15]. Common strategies include transforming complex models into simpler ones (e.g. transforming neural networks to linear models or decision trees) through global and local proximation [10, 25], visualizing the prediction result [7, 15, 26, 31], or decomposing predictions to relevance scores for model inputs [4], etc. A variety of ML interpretability methods have also been experimented with in different real-world situations, including health care and finance. Other work explores how to communicate bias and model fairness [8].

The HCI community has explored interactivity and learnability for designing visualizations that better support interpretability (e.g., [1, 2, 22]). Cheng et al. studied how different explanation strategies (e.g., "black-box" versus "white-box", and "interactive" versus "static") affected novice stakeholders' understandings of the model's decision in an algorithm-assisted college admission scenario. Elzen et al. developed a system for interactive construction and analysis of decision trees, which enabled domain experts to understand the inner workings of the algorithm and to apply their domain specific knowledge to its optimization [12]. In industry, companies such as Google and Facebook have also built visualization tools that provide support for model interpretability in running ML experiments. For example, Captum Insights[6] is an interactive visualization tool built along with the Captum interpretability library on PyTorch that helps ML engineers understand feature attribution behind individual model predictions. Although many of those tools provide effective support on interpretability in AI, most of them are applied directly on ML models for ML engineers thus there is a lack of support for application designers who are non-experts working with machine learning services (mentioned in 2.1).

2.2.2 Explaining Performance and Outcome. Other explainable AI research has been conducted around explaining a model's performance, trade-offs, and fairness based on model outcomes. Mitchell et al. [23] proposed Model Cards, which focus on trained model characteristics and inform users about what machine learning systems can do, related errors, and actions towards more fair and inclusive outcomes, which accompany machine learning models that have been released into the public domain. A recent interview study with pathologists about a diagnostic AI assistant found that users also wanted to know the design objectives of the AI systems and the "inherent trade-offs that the designers of the intelligent systems must navigate in implementing the system" ([6]). Yu et al. [37] proposed a general two-step method to help designers and users explore algorithmic trade-offs: (1) given a set of design objectives (and corresponding system criteria), generate a family of prediction models with a wide spectrum of trade-offs; and (2) create interactive interfaces to visualize the trade-offs.

In both the popular and academic press, the potential for ML systems to amplify social inequities and unfairness is receiving increasing attention [18]. Using loan granting scenarios, Wattenberg et al. [29] developed an interactive interface that shows how classifiers could potentially be unfair and also demonstrated potential strategies to turn an unfair classifier into a fairer classifier. Cabrera et al [5] created FairViz, a mix-initiative system that allows data scientists to explore intersectional bias in their models across both suggested and user-specified subgroups. Model probing platforms, such as Google's WIT, have also adopted several bias detection and mitigation features: e.g., calculating ML fairness metrics on trained models and applying fairness optimization strategies [30].

However, prior work in explainable AI has tended to primarily design, study and evaluate techniques and approaches with expert and novice users; it is not clear how the method and interfaces would help designers and developers in real-world design and development scenarios.

3 RESEARCH GAP

Prior research has demonstrated a growing need to assist application designers and ML novices in understanding trade-offs in ML systems trade-offs and in making better design decisions in

[6] Captum Insights: https://captum.ai/docs/captum_insights

building ML products. A number of interpretability methods and XAI design guidelines have been created to support this understanding. One research gap we have identified is exploring how visualization techniques could further help application designers and ML novices understand model trade-offs and fairness. We also see a need to directly deploy and evaluate these types of visualizations with Machine Learning APIs in a real-world context. In this paper, we address this gap through the design and evaluation of a set of interactive visualizations that assist application designers and ML-novices designing with the ORES ML system in Wikipedia. Our findings also contributed to broader discussions around the utilization of interactive visualization in the explainable AI domain.

4 ORES: MACHINE LEARNING SERVICE IN WIKIPEDIA

ORES is an algorithmic scoring service that supports real-time access to machine learning classifiers that predict useful characteristics of wiki edits by using multiple independent classifiers trained on different datasets [16]. The service is maintained by the Scoring Platform Team at the Wikimedia Foundation[7], the non-profit that supports Wikipedia's technical, legal, and fiscal infrastructure. The Scoring Platform Team works with wiki communities (including various languages of Wikipedia, Wiktionary, Wikidata, and other Wikimedia supported wikis) to identify opportunities in which users can apply machine learning to support wiki work processes and they develop and evaluate classifiers in collaboration with community members.

For the purposes of this paper, we focus on two specific types of classifiers available in ORES: the *damaging* and *goodfaith* models. These classifiers use a gradient boosting strategy to learn from examples of edits as labeled by Wikipedia editors, and they are commonly used in quality control processes like counter-vandalism [13]. The *damaging* model is trained to find and highlight problematic edits for review, while the *goodfaith* model is trained to find intentional vandalism and to help reviewers distinguish these edits from good-faith mistakes.

ORES's primary interface is a RESTful API that provides access to the models and generates a score for any edit in the history of Wikipedia as needed. The interface also provides access to the raw feature values that ORES uses to make predictions, the fitness statistics derived while testing the model, and even allows for the injection of synthetic feature values (aka counter-factuals) for inspection and experimentation. See [16] for an overview of these features.

ORES is used widely by volunteer application designers and professional product teams at the Wikimedia Foundation to design intelligent user interfaces and robots for editing Wikipedia's pages directly. As of September of 2020, the documentation lists over 30 different interfaces, robots, and secondary data services that use ORES[8]. Examples include the popular Recent Changes page[9] and Huggle tool[10], which allow users to review and revert recent revisions in Wikipedia articles. These tools use ORES to filter and flag suspected damaging and bad-faith edits that need more focus and allow users to revert problematic edits quickly.

[7]https://mediawiki.org/wiki/Wikimedia_Scoring_Platform_team
[8]https://www.mediawiki.org/wiki/ORES/Applications
[9]https://en.wikipedia.org/wiki/Special:RecentChanges
[10]https://en.wikipedia.org/wiki/Wikipedia:Huggle

5 METHOD & DESIGN PROCESS

We adopted an iterative, user-centered design process, with three phases: User Research, Design Objectives Gathering, and Iterative Design & Development.

5.1 User Research

We conducted interviews to understand the ORES ecosystem, and the existing pain-points and challenges that application designers face when using the ORES API. We also aimed to identify opportunities to improve participants' experiences. We recruited six people who had previously worked with ORES API by using the public research mailing list: wiki-research-l@lists.wikimedia.org. Our final interview sample consists of three ORES-based application designers, one ORES creator, one ORES core developer, and one researcher who developed research tools using ORES. Each interviewee was compensated with a $20 gift card for an approximately 45-minute interview. The semi-structured interview had a list of predefined questions along with open-ended discussion topics to help define the challenges that participants–or the application designers that they know–faced in the development process. Based on the interviews, we surfaced common themes, derived insights, and identified design opportunities.

5.2 Design Objectives

Analysis of our interview data led to the creation of four design objectives that outlined our goals in developing the ORES Explorer visualization system:

- ***Explain digestible ML concepts in the context of Wikipedia.*** Most participants mentioned that they did not have a Machine Learning background prior to working with ORES and thus struggled to understand ML concepts when reading the documentation. For example, when asked about background, one participant answered that *"everything that I know about AI and ML is through working with ORES. What I do is that I have to look things up over and over"*. Although the ORES documentation does provide links to Wiki pages for most of the terminologies used, it also forces application designers to make and constantly go back and forth between different Wiki pages. To solve this problem, ORES Explorer could provide some basic explanation on the core ML concepts within the context of Wikipedia. This explanation could help reduce the barriers and confusion for the application designers before they make important decisions using the ORES models.
- ***Facilitate understanding of trade-offs in threshold setting.*** Choosing the right threshold is the key to effectively using the ORES predictions. To do that, application designers need to understand the algorithmic trade-offs and make informed decisions based on the context. For example, a low threshold ensures that the majority of damage is caught at the cost of needing to review more edits. Conversely, a high threshold minimizes the harm of automatically reverting good edits but might let a large number of damaging edits

pass by[11]. During our interviews, several participants expressed concerns over choosing the appropriate threshold for their tools. One participant, for example, found some guidance on the ORES Wiki page, but later found out that the result was not as promising and sufficient as she had expected. She wished that there were more resources for her to dig into and explore herself. We therefore wanted to provide a way for users to explore different threshold settings and intuitively understand how those settings will affect the different values they care about.

- ***Suggest threshold settings based on use cases and design goals.*** Application designers use ORES for different reasons and to build different types of tools. For example, some tools are built to flag damaging edits for the human reviewers while others (automated bots) can directly revert them. The purposes of the tool can directly affect how application designers should set the thresholds. For example, a higher threshold should be used on the automated bots in order to catch the most damaging edits while avoiding misclassifications of the good edits. However, some application designers just blindly go with "50%" since it looks like the default option. Thus, we believe that suggested thresholds should be given as a reference based on application designers' goals and the type of tools that they are building.
- ***Explain model performance disparity in editor groups.*** Prior work [16, 28] demonstrates that there is a disparity in ORES performance on edits from different editor groups. For example, the model is more aggressive to edits from newcomers and anonymous editors. Thus, it's more likely for good edits from these two groups to be misclassified as damaging edits (as compared to those edits from the experienced, logged-in editors). During the interview, several participants mentioned that they had noticed or heard of such a disparity in their previous experience of working with ORES but had difficulty further exploring the disparity. For example, one participant mentioned that it would be helpful to access any information about the potential issues before using the ORES API. We would, therefore, like to provide ways for people to explore and recognize model disparity, which would be helpful in driving important product feature decisions.

5.3 Iterative Design and Development

We generated different design concepts that met our design objectives, ultimately developing an interactive visualization system based on feedback from informal discussion with Wikipedia domain experts. We further refined the final concept, starting with low-fidelity mock-ups and moving to high-fidelity designs, using a similar iterative process. We developed the front-end visualization website in React, iteratively coding by connecting to a sample dataset of 500 Wikipedia edits with ORES prediction scores. This process resulted in the development of a visualization tool with fully interactive features that were published as a functional website.

5.3.1 Data. The sample data we used in the visualizations consists of 500 Wikipedia edits randomly selected from the training data of the English ORES model. To avoid over-fitting and showing untrue performance metrics, we constructed a replicate machine learning model of ORES, using exactly the same machine learning algorithm (Gradient Boosting), label weights (10 for damaging edits, 1 for non-damaging edits) and hyper-parameters, while excluding the 500 edits we selected from the training data. Then, we applied this replicated ORES prediction model to the 500 edits on which the model was not trained, to generate out-of-bag prediction results and performance.

For the purpose of visualizing model performances in groups, we selected a balanced number of newcomer and experienced editors' edits from the 500 edit dataset. Within each editor group, we also balanced the number of damaging and non-damaging edits. We balanced the data with respect to the types of editors and the number of damaging and non-damaging edits to make visualizations more illustrative. For example, 96% of the data in the original training dataset is labeled as non-damaging, therefore showing the distribution with the original ratio would make the damaging edits extremely difficult to identify. Having a balanced data set on types of editors and within-group damaging rates allows users to make clear comparisons between different editor groups, as shown in Figure 4.

6 ORES EXPLORER SYSTEM

ORES Explorer is an interactive visualization website that allows user to explore and understand the algorithmic trade-offs and model fairness in ORES via a sample dataset. The visualization website was designed to meet our design objectives through the combination of four individual visualizations:

- **About ORES** Explaining ORES API and necessary ML knowledge specific to Wikipedia
- **Threshold Explorer** Visualizing trade-offs in setting different performance threshold
- **Threshold Recommender** Recommending thresholds based on the application types
- **Group Disparity Visualizer** Communicating model performance disparity in editor groups

6.1 Landing Page: About ORES

About ORES serves as a landing page that provides a basic overview of the ORES system. The goal of the About ORES section is to provide the necessary knowledge for users to explore the following pages. Visualizations are used to explain concepts such as how ORES scores edits, and how ORES makes predictions based on threshold settings and the prediction confusion matrix in the context of Wikipedia.

6.2 Threshold Explorer

The Threshold Explorer is the interactive visualization that allows users to explore impact in model performance by setting different thresholds. Users will first choose a model and slide the top slider to perceive changes in the model's prediction and performance. The graph underneath the slider directly represents individual edits and their prediction results distributed from left to right based on their prediction scores.

[11] https://www.mediawiki.org/wiki/ORES/Thresholds

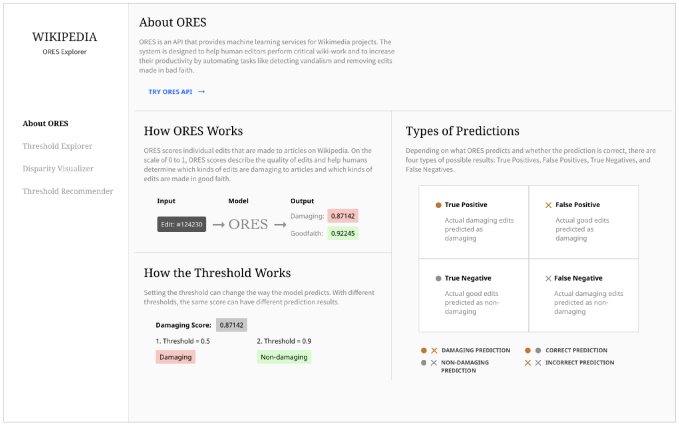

Figure 1: Landing Page: About ORES. The landing page includes several sections: 1) general information on the ORES system 2) ORES's process of making predictions 3) How threshold works in the ORES system 4) four types of possible prediction results

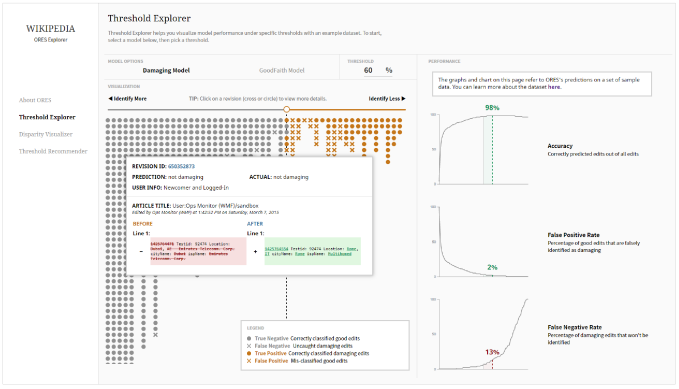

Figure 2: Threshold Explorer. In the center, there is an interactive visual field for users to explore different thresholds by sliding the threshold slider. The right side is the performance panel that shows different performance metrics under a specific threshold setting

Based on the confusion matrix, two colors (grey and orange) were used to represent the predicted classes, and two shapes (a circle and cross) were used to represent the accurate/inaccurate predictions. When a user interacts with the threshold slider, they perceive changes in the circles and crosses to quickly gain a visual understanding of how a threshold is affecting the model performance. For example, when the slider is moved from left to right, the user will see increasing orange crosses (false-positive predictions) and decreasing grey crosses (false-negative predictions). Additionally, each edit in the sample data set represented by circles and crosses is clickable so that users can see detailed information about the edit and its prediction result.

On the right side of the visualization, the performance panel provides quantitative performance information of different threshold settings. With the user's interaction on the threshold slider, the percentage number and charts on Accuracy, False Positive Rate, and False Negative Rate change accordingly. Users can easily perceive all three performance metrics for a particular threshold or compare performance metrics among thresholds. A line chart also helps users understand the relationship between a given performance metric and the threshold setting. For example, the line chart of the False Positive Rate indicates that there is an inverse relationship between the threshold and the False Positive Rate. The three metrics chart is designed to make it easier for people to prioritize what values they care about and eventually select a threshold that is aligned with their design goals.

6.3 Threshold Recommender

The Threshold Recommender presents a different approach to help the user choose the desired threshold. Instead of allowing the user to explore different thresholds, the Recommender asks the user to choose an overall goal for their applications and then it directly recommends a threshold to use. The interactive visualization illustrates three options using visualizations and descriptions. Once a user selects a goal, the system will provide a threshold suggestion and allow them to adjust the suggestion according to their preferences. In the suggestion panel, the user can also directly see the impact of the particular threshold and request full model performance.

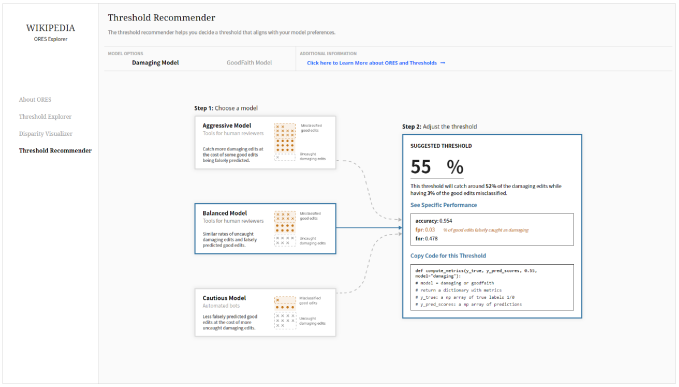

Figure 3: Threshold Recommender. There are three predefined types of model configuration for users to choose from base on their use case. After the user chooses one card from the left, the card on the right will suggest a threshold for users to further adjust and explore the model performance.

6.4 Group Disparity Visualizer

Prior work suggests that there is a disparity in false positive and false negative rates between edits from different editor groups, e.g. newcomer and experienced editors. The Group Disparity Visualizer helps to surface those fairness issues by allowing users to see and compare the model's performance on different groups of edits using a sample dataset. It offers two ways of categorizing the edits: by the editor's experience level and by the edit's anonymity.

The visualization displays the total number of edits in the four prediction categories (true positive, true negative, false positive, false negative) for each edit group and a side panel that shows the disparity in accuracy, false-positive rate, and false-negative rate between the two groups. To maintain consistency, the visualization uses the same color-coding methodology as in the Threshold Explorer. Threshold slider and model selection are also included to help users observe group disparity under different thresholds and models. The side panel enables the users to statistically compare the model's performance in the two groups. For example, using the damaging model with a threshold of 60%, the false positive rate for edits from newcomer is 6.3% while and false positive rate for the experienced editors is 0%. By interacting with the Group Disparity Visualizer, users will be able to evaluate the model from a fairness perspective and potentially take action to address the disparity in model performance.

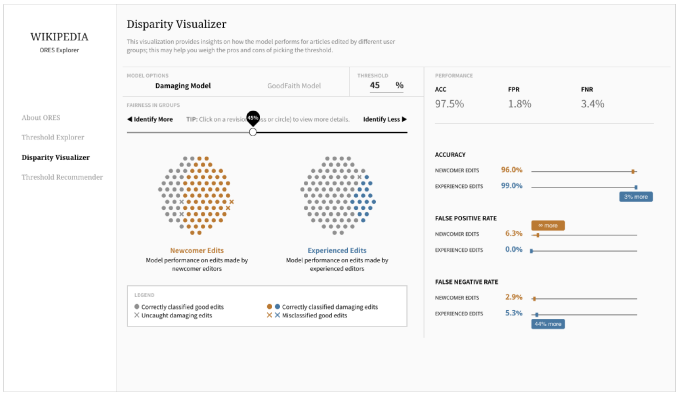

Figure 4: Group Disparity Visualizer. On the middle-left side, the interactive visual field allows users to slide the threshold to visually explore model performance disparity between different editor groups. On the left, the performance panel shows statistic information about the model performance in different groups.

7 EVALUATION

In creating our evaluation for the ORES Explorer, we sought to answer three questions that corresponded to our design objectives:

(1) How will the visualization tool meet our design objectives?
(2) How does the tool change application designers' perceptions around AI systems?
(3) How can the tool be effective for people less familiar with the Wikipedia context?

7.1 Method

To answer these questions, we designed a think-aloud interview protocol incorporating a design task where we asked participants to first choose an ORES-based tool that they would like to build in a real-world environment. We provided them two options: a review tool that flags edits to a human reviewer or a bot that automatically reverts edits. Participants then used our visualizations to choose a threshold for the application that they had chosen while learning about model performance disparity in editor groups. While participants explored each interface, we observed how they were interacting with the features and asked them to vocalize their thought processes out loud. After the design task, we asked participants follow-up questions related to their understanding of the trade-offs and fairness within the ORES model, their perceptions on the ORES AI system, and their understanding of the ML concepts in the Wikipedia context. Using this protocol, we thoroughly evaluated our visualization and gained insights on how participants learned about the ML system and understood model trade-offs through the visualization system for making sensible product decisions in designing ML applications.

7.2 Participants

For our study, we recruited 10 participants from two groups of application designers, as potential users of our visualization system. Prior to the study, none of the participants had any substantial machine learning knowledge, thus their participation could help us learn about how ORES Explorer can help ML novices understand the model's trade-offs when designing ORES-based applications. The first group included five application designers from the Wikipedia community (ORES application designers and researchers), recruited through Wikipedia's technical Village Pump forum and the Wikipedia research mailing lists. We recruited them to help us understand how current Wikipedia stakeholders (and potential direct users of the visualization tool) might interact with our visualization and understand the ORES model's trade-offs. Smith et al. [27] found that Wikipedia stakeholders share common community values with regard to the use of Machine Learning-based systems. The second group included five application designers outside of the Wikipedia community, recruited through snowball sampling that was focused on networks within the tech industry. All selected participants had a basic understanding of AI and how Wikipedia works but did not have a traditional ML or stats background. This second group was, therefore, recruited to help us evaluate how well the visualizations would be understood with potentially new application designers unfamiliar with the Wikipedia context. Table 1 contains voluntarily disclosed participant information.

Participants joined the evaluation voluntarily and were compensated with a $20 gift card for an approximately 45-minute long interview. We explained the context and goals of the study, and gave participants time to explore each of the four interfaces. Because our participants were located around the world, we used zoom.us to conduct interviews, asking participants for permission to share and record their screens as they used the visualization tools.

7.3 Analysis

We followed Charmaz's approach to grounded theory [9] for our data collection and analysis. Two members transcribed each interview and open coded the transcripts, and our whole team met to discuss the themes that emerged in the open coding. Themes emerged organically from the analytic process and were then supplemented by existing work. The themes were then summarized into seven key findings presented in the next section.

Table 1: Participant Information.

ID	Group	Role
P1	Outside Wikipedia	Associate UX Designer, Spotify
P2	Outside Wikipedia	Product Designer, Facebook
P3	Outside Wikipedia	Digital Product Manager, CitiGroup
P4	Outside Wikipedia	Student, Harvard Graduate School of Design
P5	Outside Wikipedia	UX Designer, Amazon
W1	Wikipedia Member	Edit Reviewer, Recent Changes Tool User
W2	Wikipedia Member	Volunteer Editor, Implemented ORES at eu.wikipedia.org
W3	Wikipedia Member	Developer, Researcher, Volunteer for Wikimedia Projects
W4	Wikipedia Member	Researcher, Edit Reviewer, Recent Changes Tool User
W5	Wikipedia Member	Huggle Application Designer

8 RESULTS

Our participants found ORES Explorer to be useful in educating them about ML concepts and in helping them to understanding trade-offs and performance disparities in ML models. We organized our findings into the following three themes, along with several key findings:

Meeting design objectives on educating contextualized ML concepts, threshold trade-offs and model performance disparity

- Contextualized ML concepts in the About ORES page were helpful for participants with limited ML expertise to start learning about the ORES API.
- ORES Explorer improved participants' understanding of the trade-offs in setting different ORES model thresholds and the associated impact on the tool they chose to build in the design task.
- Participants were able to use Threshold Explorer and Threshold Recommender to make decisions and evaluate trade-offs about desired thresholds that aligned with their goals.
- Although Group Disparity Visualizer helped surface the ORES model's performance disparity in different editor groups, most participants accepted the disparity as a natural occurrence and were not concerned about fairness implications in the system.

Participants' perceptions on the underlying ML system

- When balancing different performance metrics to select a model threshold, participants tended to prioritize model accuracy over other metrics.
- Overall, ORES Explorer helped participants to trust the underlying AI models more. The visualizations surfaced AI models' limitations and the resulting transparency allowed participants to see the AI models not as "black boxes," but rather as helpful and editable tools.

Effectiveness of the visualization tools for users with different backgrounds

- ORES Explorer was shown to be effective in helping participants understand the ORES ML system without Wikipedia domain background.

In the following section, we discuss our findings in detail.

8.1 Assist contextual learning of ML concepts in Wikipedia

We found that learning about ML concepts in the context of Wikipedia is essential for application designers to start designing with the ORES API. Participants mentioned that, at the beginning of the design task, they were challenged to understand what different performance metrics–such as False Negatives and False Positives–meant in order to learn about the algorithmic trade-offs and model disparity. In the group outside of Wikipedia, P4 commented *"I think the difficult part is to understand what exactly is a false and positive means in a certain scenario"* (P4). From the group inside Wikipedia, W1 commented *"I always forget the meanings of the false positive rate and false negative rate. I always have to interpret in the context of the model"* (W1). Participants thought that the one-sentence explanation (underneath each concept), as well as the use of iconography (circles and crosses) were helpful in facilitating the contextual understanding of the confusion matrix. However, participants who had no prior experience with ML were sometimes confused, mixing up the meanings of false positives and false negatives. *"I think decreasing false positives might be the most important ... Actually no, sorry, let me figure out what the terms are. It always gets so confusing ... Okay, no, yeah, I'm gonna say percentage of false negatives is more important to be lower..."* (P5).

8.2 Improve overall understanding on model threshold trade-offs

We observed that participants gradually gained knowledge about the ORES model threshold trade-offs by interacting with the Threshold Explorer. As participants moved the threshold slider and perceived changes, they intuitively understood the underlying trade-offs among key model performance metrics in the context of Wikipedia. *"I see, the lower the threshold goes, a lot more edits sneak by [false positives] … But at the same time, that also lets more good-faith edits through without being flagged [true negatives]"* (W5). Along with the threshold slider, the performance charts on the performance panel also helped surface trade-offs, specifically for accuracy, false positive rate, and false negative rate. All of the participants in our study based their decisions, at least partially, on the performance charts, and seven out of ten participants reported primarily focusing on them. As P5 commented, *"false positives and false negatives are pretty clearly inversely related to each other [as] you can see from the shape of the graphs, whereas accuracy sort of has a sweet spot."*

Furthermore, Threshold Explorer helped participants understand how to adjust the model threshold in order to prioritize different performance metrics, which correspond with different community values. For example, all participants increased the threshold when asked to minimize misclassifying good edits in Wikipedia. W4 indicated *"I'll have to play with the false positive rate. Okay, when I move this way it's reducing good edits that are falsely identified as damaging. So then I need to move this to the right."* The model trade-offs that were shown are valuable for making model threshold decisions.

8.3 Enable participants to select thresholds based on their design goals

The Threshold Recommender facilitated participants' decision making in selecting thresholds based on their application design goals. The average damaging threshold among the seven participants who design review tools is 0.48 (standard deviation = 0.09), while the average damaging threshold among the three participants who design automated bots is 0.61 (standard deviation = 0.08). Commenting on his experience choosing a threshold to build an automated bot, W2 said, *"[If I choose a] small threshold we have lots of good faith and not damaging edits that are classified as damaging so that is quite annoying especially if we are making a bot that will revert automatically."* Thus, some participants found Threshold Recommender useful as it directly suggested thresholds that were aligned with their design goals. *"I can just choose one type of model based on what I want"* (P1).

While some participants found that Threshold Recommender could be more practical and require less cognitive workload compared to Threshold Explorer, others also found it less educational in facilitating their understanding of the underlying relationships among different model performance metrics. *"I think the first one (Threshold Explorer) explains the relationship better while I'm playing with the visualization and looking at the graphs on the right. I think like the first one is more educational versus this one (Threshold Recommender) is more practical"* (P1).

On the other hand, the threshold suggestion from Threshold Recommender could also lead participants to question the logic behind those direct suggestions. Some participants conveyed that they found the suggestion less trustworthy due to the lack of explainability. *"I don't know why there's a default suggested number... and some explanation on how the suggestions are generated will be helpful"* (P2).

8.4 Participants attributed the model performance disparity to a natural occurrence

The Group Disparity Visualizer helped to surface how the model could perform differently among different editor groups. By interacting with this visualization, eight out of ten participants found it intuitive to understand the model performance disparity. *"I just feel like the system treats these two user groups differently. Definitely obvious, it is more aggressive to newcomers and more gentle to the experienced edits"* (P4). However, nine out of ten participants also paid more attention to the editing ability of the two editors groups instead of the model's performances on the two editors groups.

While participants perceived the performance disparity and potential fairness problem within the model, 7 out of 10 participants (4 out of 5 Wikipedia participants, and 3 out of 5 non-Wikipedia participants) attributed the performance disparity to a natural occurrence, and were, therefore, not particularly concerned about potential fairness problems. For example, P3 commented that *"this makes sense, because for experienced edits, the machine already has experience looking at their edits, so it has a higher accuracy"* (P3). From the group inside Wikipedia, W1 commented that *"it's probably to be expected, because for experienced users, they will likely have more examples of edits to train the model... maybe it's over-fitting to experienced editors' edits, [but] I don't think it's necessarily bad because we want more edits which look like it's made by experienced users"* (W1).

8.5 Participants prioritized accuracy when balancing different performance metrics

Based on the understanding of model threshold trade-offs, most participants tried to strike a balance between multiple evaluation metrics for the model in the evaluation design task. At first glance, P2 noted: *"Okay, I guess the goal here is to like, find the balance, where we have a good number of accuracy and like, not too low on these two as well."* Seven out of ten participants mentioned that they would keep a high accuracy before considering other factors. *"I'm trying to maximize accuracy, and at the same time lower false-positive rate or false-negative rate"*(P1). Some participants also decided to compromise on certain performance metrics in order to achieve particular design goals. W5 commented *"It's better to have a false positive than a false negative … a lot of the times false negatives will sneak through and they'll sit in an article for weeks or months before somebody goes, "Hey, this doesn't sound right", and actually goes about changing it."*

8.6 Trade-offs education and transparency enhanced overall user trust

After interacting with ORES Explorer, eight out of ten participants indicated that they had gained more trust in using the machine learning model. With limited machine learning expertise, some

participants mentioned that the transparency that the visualizations provided them helped them to understand how the model actually works so that they were no longer perceiving it as only a "black box" capability. *"Now that I can visualize what is actually happening with this data set, I could just tell that it actually does work. And it's not just voodoo magic occurring behind the API"* (W5). *"I definitely feel like seeing this type of visualization helps me trust AI more in the sense that I understand how much of it is actually due to human decision making like, because the AI model is really just like complicated statistics and it's the humans who decide what matters on which Just kind of an opinion that I already have"* (P5).

Participants also talked about how the experience of having guidance when making customized adjustments to the model threshold contributed to a sense of "shared responsibility" with the machine learning model. *"I felt I have a shared responsibility with the model to prioritize which kind of things I will get right and which I will get wrong, because I cannot have both at the same time in this. Let's say this slide here provides something like that I can maximize getting one thing and I have to give up on the other aspect"* (W1). *"I like the way that it's customizable. So I feel like it's not just the AI working on itself."* (P3).

8.7 The visualization tools are shown to be effective for participants without Wikipedia domain expertise

When comparing the interview sessions between the two groups of participants recruited inside and outside Wikipedia, we found no distinct difference in how the two groups perceived and interacted with ORES Explorer. Based on our recording of the thresholds that participants chose in the design task, the group of participants outside Wikipedia on average tended to choose moderately more aggressive prediction thresholds as compared to the Wikipedia participants. The Wikipedia group chose an average threshold of 0.58 (standard deviation = 0.08), while the group outside of Wikipedia chose an average threshold of 0.46 (standard deviation = 0.1). The group outside of Wikipedia also recognized how the algorithmic trade-offs would affect Wikipedia's community values. *"I'm gonna say the percentage of false negatives is more important to be lower in this case (building a review tool).… too many false negatives will really downgrade the quality of the system for the readers"* (P5). These results showed that ORES Explorer can be helpful for users from different backgrounds—editors, developers, product manager and designers—or users with limited application domain knowledge.

In evaluating why participants outside of Wikipedia chose more aggressive prediction thresholds, one explanation is that they might tolerate higher false-positives (flagging good edits as damaging) since they have not done any editing work or worked with Wikipedia editors before. While participants inside Wikipedia, by comparison, tend to tolerate fewer false positives, they tend to be more familiar with the reviewing process and express more concern that having more false positives will overwhelm the reviewers shouldering review work. W5 comments that one *"difficulty with setting a threshold is that you don't want to get so many false positives that the reviewer starts getting fatigued and just will not use that tool anymore."* Further research evaluating how users' backgrounds affect the ways that they balance the algorithmic trade-offs when designing with machine learning services would be worthwhile.

9 DISCUSSION

Our research investigated two significant challenges faced by ML novice designers when designing ML applications: deciding on model thresholds, and understanding model fairness. When building most AI products with classification tasks, setting the appropriate model decision threshold is a crucial step in determining how the product will weigh false positives and false negatives, which can have lasting impacts on both user experiences and societal outcomes in some high-stake industries. On the other hand, understanding model fairness in terms of a model's performance among different social groups is also critical in helping designers make informed decisions on model usage and in helping them design responsible product features.

While ORES Explorer was designed and evaluated in the real world scenario of the Wikipedia ORES system, we believe that our visualization framework is readily generalizable in other systems and could readily be applied in different use cases. Our evaluation revealed that even participants without Wikipedia background gained a similar understanding of model trade-offs as compared to Wikipedia participants. In other online communities such as Facebook and Reddit, similar content moderation models have been widely applied in order to regulate fraudulent or dangerous content. Beyond content moderation, other high-stake AI use cases, such as criminal sentencing or disease diagnosis, are also in need of tools to help application designers and stakeholders understand model trade-offs and fairness in order to make model threshold decisions and assess model limitations. Interactive features from our work, including threshold sliders, confusion matrix dot maps, and group performance visualizers, would allow application designers working in these contexts to effectively explore model trade-offs and fairness, to make decisions about model thresholds, or to compare models in order to minimize ML harm. One line of future work is to build on ORES Explorer to create a model-agnostic visualization tool that allows users to visualize model value trade-offs and fairness with custom models and use cases. Further research and evaluation around a model-agnostic tool could potentially and systematically improve transparency for designers using ML models.

In order to communicate model threshold trade-offs, we explored and evaluated two specific visualization methods. Threshold Explorer visualizes the changes to a model's accuracy, false-positive rate and false negative rate when users interact with the threshold slider on a sample dataset. Threshold Recommender took a different approach, directly suggesting threshold values based on a pre-defined set of design goals. In our study, we found that the two visualizations are complementary in facilitating an application builder's understanding of model trade-offs. While some participants commented that direct recommendation (Threshold Recommender) is more practical and efficient, others mentioned that it is less educational and less transparent when compared to the interactive exploration on the sample data set (Threshold Explorer). It would be worthwhile, in future work, to thoroughly evaluate these visualization methods, along with other visualization techniques,

to provide guidelines for how those methods can be adopted in different contexts.

While our evaluation provided evidence that our visualization is effective in communicating model threshold trade-offs in ORES, there are some limitations in this work that leave room for improvement and more design exploration in the future. We found that it was difficult to separate the participants' superficial understanding of model fitness from suitability. For example, when setting the model threshold, seven out of ten participants chose to prioritize the accuracy measure. This is surprising, because many workflows that use ORES in Wikipedia[27] lend themselves better to measures like precision, recall, and false positive rate. Accuracy can be a misleading and less relevant metric, due to the typically imbalanced ratio between damaging and non-damaging edits. Thus, an additional topic for future research is to explore how we could leverage design to further explain ML concepts, such as accuracy, in a more familiar way.

The present research points to the need for a broader conversation about how ML novices perceive model fairness. When interacting with the Group Disparity Visualizer, some novice participants successfully perceived that the model was "treating" edits from different editor groups differently. However, most participants believed that the model should be more aggressive to newcomers because they are more likely to make mistakes. These participants were likely confused about editor performance (the quality of the editor's edits) and model performance (model's ability to correctly classify the edits), and thus unable to correctly evaluate the model's fairness. In future work, we could explore how to better help people distinguish between these two issues, and how to develop empathy for the disadvantaged groups such as, in this case, the newcomer editors. For example, we could provide the model predictions for a given editor's first few edits and let application designers see how their edits would be treated from a newcomer's perspective.

One interesting finding from this work was that participants reported developing greater trust towards the ORES system after interacting with the visualizations. As we previously mentioned in the results, the process of enabling application designers to explore the ORES model gave them a sense of "co-creation" and "shared responsibility." Participants also commented that the design of visual artifact made the complex ML concepts more "approachable" and "intuitive to process". This opens up a new opportunity to explore how to leverage design to better establish trust in AI systems. In many other real world scenarios, such as social media content moderation, criminal sentencing and disease diagnosis, the aspect of trust is essential to build more effective human-AI collaboration. Additionally, there is a potential to expand on the concept of "co-creation" and design in order to engage ORES users in the actual model development process. Wikimedia Foundation has done some related work in the area of collaborative AI development. Jade[12] is a MediaWiki extension that is designed to allow editors to annotate articles, revisions, diffs, and other Wikipedia components. It offers a simple workflow to collect human labels from the Wikipedia community and to provide data to train better versions of the ML models. In the future, we could explore ways to build on the transparency provided by our visualization tool with user feedback mechanisms that could help improve our models overtime and further enhance user trust towards the overall ML system.

[12]Jade: https://www.mediawiki.org/wiki/JADE

10 CONCLUSION

In this study, we presented the design, development, and evaluation of ORES Explorer, an interactive visualization tool to communicate the algorithmic trade-offs and model performance disparity in ORES, the ML service for editing moderation in Wikipedia. We reported on a case study conducting in-depth interviews with current and potential application designers in and outside the Wikipedia community to evaluate our visualization. The study results demonstrated that our system is helpful in facilitating designers and developers with limited ML knowledge to develop a better understanding of the ML system and make decisions that align with their design goals. Our study provided future opportunities for the design community in exploring general visualization techniques to provide greater transparency when designing with AI systems.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation (NSF) under Award No. 2001851, 2000782 and 1952085, and the NSF Program on Fairness in AI in collaboration with Amazon under Award No. 1939606. We also thank Xingyu Liu, Hwi Kyung Nam, Zhuona Ma, and Dr. Vincent Aleven for their input and feedback to the work.

REFERENCES

[1] Ashraf Abdul, Jo Vermeulen, Danding Wang, Brian Y Lim, and Mohan Kankanhalli. 2018. Trends and trajectories for explainable, accountable and intelligible systems: An hci research agenda. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–18.

[2] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *Ai Magazine* 35, 4 (2014), 105–120.

[3] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. Machine bias. *ProPublica, May* 23 (2016), 2016.

[4] Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. 2016. Layer-wise relevance propagation for neural networks with local renormalization layers. In *International Conference on Artificial Neural Networks*. Springer, 63–71.

[5] Ángel Alexander Cabrera, Will Epperson, Fred Hohman, Minsuk Kahng, Jamie Morgenstern, and Duen Horng Chau. 2019. FairVis: Visual analytics for discovering intersectional bias in machine learning. In *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 46–56.

[6] Carrie J Cai, Samantha Winter, David Steiner, Lauren Wilcox, and Michael Terry. 2019. " Hello AI": Uncovering the Onboarding Needs of Medical Practitioners for Human-AI Collaborative Decision-Making. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–24.

[7] Rich Caruana, Scott Lundberg, Marco Tulio Ribeiro, Harsha Nori, and Samuel Jenkins. 2020. Intelligible and Explainable Machine Learning: Best Practices and Practical Challenges *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 3511–3512. https://doi.org/10.1145/3394486.3406707

[8] Rich Caruana, Scott Lundberg, Marco Tulio Ribeiro, Harsha Nori, and Samuel Jenkins. 2020. Intelligible and Explainable Machine Learning: Best Practices and Practical Challenges. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3511–3512.

[9] Kathy Charmaz. 2014. *Constructing grounded theory*. sage.

[10] Mark Craven and Jude Shavlik. 1999. Rule extraction: Where do we go from here. *University of Wisconsin Machine Learning Research Group working Paper* 99 (1999).

[11] Graham Dove, Kim Halskov, Jodi Forlizzi, and John Zimmerman. 2017. Ux design innovation: Challenges for working with machine learning as a design material. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 278–288.

[12] S. V. D. Elzen and J. V. Wijk. 2011. BaobabView: Interactive construction and analysis of decision trees. *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)* (2011), 151–160.

[13] R Stuart Geiger and Aaron Halfaker. 2013. When the levee breaks: without bots, what happens to Wikipedia's quality control processes?. In *Proceedings of the 9th International Symposium on Open Collaboration*. 1–6.

[14] Marco Gillies, Rebecca Fiebrink, Atau Tanaka, Jérémie Garcia, Frédéric Bevilacqua, Alexis Heloir, Fabrizio Nunnari, Wendy Mackay, Saleema Amershi, Bongshin Lee, et al. 2016. Human-centred machine learning. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. 3558–3565.

[15] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2018. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*. IEEE, 80–89.

[16] Aaron Halfaker and R Stuart Geiger. 2019. ORES: Lowering Barriers with Participatory Machine Learning in Wikipedia. *arXiv preprint arXiv:1909.05189* (2019).

[17] Lars Erik Holmquist. 2017. Intelligence on tap: artificial intelligence as a new design material. *interactions* 24, 4 (2017), 28–33.

[18] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé III, Miro Dudik, and Hanna Wallach. 2019. Improving fairness in machine learning systems: What do industry practitioners need?. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–16.

[19] Harmanpreet Kaur, Harsha Nori, Samuel Jenkins, Rich Caruana, Hanna Wallach, and Jennifer Wortman Vaughan. 2020. Interpreting Interpretability: Understanding Data Scientists' Use of Interpretability Tools for Machine Learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–14.

[20] Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. 2016. How we analyzed the COMPAS recidivism algorithm. *ProPublica (5 2016)* 9, 1 (2016).

[21] Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. 2016. *How We Analyzed the COMPAS Recidivism Algorithm.* https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm

[22] Q Vera Liao, Daniel Gruen, and Sarah Miller. 2020. Questioning the AI: Informing Design Practices for Explainable AI User Experiences. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–15.

[23] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*. 220–229.

[24] 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.

[25] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386* (2016).

[26] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 1135–1144.

[27] 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*. 1–14.

[28] Nathan TeBlunthuis, Benjamin Mako Hill, and Aaron Halfaker. 2020. The effects of algorithmic flagging on fairness: quasi-experimental evidence from Wikipedia. *arXiv preprint arXiv:2006.03121* (2020).

[29] Martin Wattenberg, Fernanda Viégas, and Moritz Hardt. 2016. Attacking discrimination with smarter machine learning. *Google Research* 17 (2016).

[30] James Wexler, Mahima Pushkarna, Tolga Bolukbasi, Martin Wattenberg, Fernanda Viégas, and Jimbo Wilson. 2019. The what-if tool: Interactive probing of machine learning models. *IEEE transactions on visualization and computer graphics* 26, 1 (2019), 56–65.

[31] Abraham J Wyner, Matthew Olson, Justin Bleich, and David Mease. 2017. Explaining the success of adaboost and random forests as interpolating classifiers. *The Journal of Machine Learning Research* 18, 1 (2017), 1558–1590.

[32] Qian Yang. 2018. Machine learning as a UX design material: How can we imagine beyond automation, recommenders, and reminders?. In *AAAI Spring Symposia*.

[33] Qian Yang, Alex Scuito, John Zimmerman, Jodi Forlizzi, and Aaron Steinfeld. 2018. Investigating how experienced UX designers effectively work with machine learning. In *Proceedings of the 2018 Designing Interactive Systems Conference*. 585–596.

[34] Qian Yang, Aaron Steinfeld, Carolyn Rosé, and John Zimmerman. 2020. Re-examining whether, why, and how human-ai interaction is uniquely difficult to design. In *Proceedings of the 2020 chi conference on human factors in computing systems*. 1–13.

[35] Qian Yang, Aaron Steinfeld, Carolyn P Rosé, and John Zimmerman. 2020. Re-examining Whether, Why, and How Human-AI Interaction Is Uniquely Difficult to Design. In *Proceedings of the 2020 chi conference on human factors in computing systems*.

[36] Qian Yang, Jina Suh, Nan-Chen Chen, and Gonzalo Ramos. 2018. Grounding interactive machine learning tool design in how non-experts actually build models. In *Proceedings of the 2018 Designing Interactive Systems Conference*. 573–584.

[37] Bowen Yu, Ye Yuan, Loren Terveen, Zhiwei Steven Wu, Jodi Forlizzi, and Haiyi Zhu. 2020. Keeping Designers in the Loop: Communicating Inherent Algorithmic Trade-offs Across Multiple Objectives. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*. 1245–1257.