

# Tinkering: A Way Towards Designing Transparent Algorithmic User Interfaces

Dilruba Showkat

Lehigh University, Bethlehem, PA, USA

## Abstract

With the widespread use of algorithms in interactive systems, it becomes inevitable for the users to apply these algorithms with caution. Algorithms are applied to make decisions in healthcare, hiring, the criminal justice system, and social media news feed among others. Thus, algorithmic systems impact human lives and society in significant ways. As a consequence, currently, the focus has been shifted toward designing transparent algorithmic user interfaces (UI's) – to make the algorithmic aspects more explicit. Designing transparent algorithmic user interfaces requires the designer to bring the algorithmic aspects of control at the UI level without causing information overload. This research attempts to investigate this gap by proposing tinkering or playful experimentation as a means of designing transparent algorithmic UI's. Tinkering is a cognitive style related to problem-solving, decision making, enables exploration with the interactive system. The proposed approach of combining tinkering with transparent UI's serves two potential purposes: first, the exploratory nature of tinkering has the ability to make the algorithmic aspects transparent without hurting users experience (UX), while providing flexibility and sufficient control in the personalized interactive experience; second, it enables the designer to detect software inclusiveness issues in the design before they become part of the final software, by allowing us to measure *how much* algorithmic transparency is desired across different user groups.

## Keywords

tinkering, exploration, transparent algorithmic user interface, inclusive design, algorithmic transparency

## 1. Research Problem and Motivation

Algorithms are rapidly applied in most of the interactive applications that we use today. For instance, well-known algorithmic system includes YouTube for video recommendation, COMPAS risk assessment tool [1], Facebook News Feed [2] among others. Research shows, that in many cases, these systems generated predictions might suffer from biases [1, 3, 4], causing accountability and safety issues [5], due to lack of clarity of the underlying mod-

els and algorithms; in worst, they might enable the users to make a wrong decision. Users trust is violated when these algorithmic systems produce an outcome that is harmful, biased, and unethical. As a consequence, sometimes users end up stop using such product or services [6, 7]. Thus, designing transparent algorithmic user interfaces are getting more and more attention among the research community [8] to make the algorithmic aspects explicit and more transparent.

Previous research have advocated for transparent recommendation systems in various domains [9, 10], transparent statistical research practices [11], transparent debugging [12, 13], and transparent journalism [14, 15]. While others have examined and emphasized the importance of transparent data collection process [16, 17]; Microsoft datasheets for datasets presents one example [18, 17] to achieve trans-

Joint Proceedings of the ACM IUI 2021 Workshops, April 13-17, 2021, College Station, USA

 dilrubashowkat@gmail.com (D. Showkat)



© 2021 Copyright © 2021 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0)



CEUR Workshop Proceedings  
(CEUR-WS.org)

parency and accountability during Machine Learning (ML) lifecycle for both dataset creator and dataset consumer.

Similarly, various techniques are also available to make the underlying algorithmic assumptions more open, interpretable, and easy to understand; explanation is one of them [2, 19]. Recently, researchers also explored the potential of socio-technically inspired perspective such as Social Transparency (ST) [20], perhaps, due to the social nature of interpretability [21]. Explanation tool or Explainers, also known as interpretability tool are available as open-source Python packages to describe both white box and black box models [18, 22, 23, 24]. They provide an easy interpretation of the model’s mechanism and outcome in a trustworthy, transparent, and safer manner [25]. Applying these explainers requires the user to call pre-defined functions, integration with complex workflows [21], and is often “critiqued for its techno-centric view” [20]; and, applying them requires programming. Furthermore, as these tools are publicly available and free to use, research shows that even expert data scientists overuse the explainers (InterpretML [22]) prediction by overly trusting them, and sometimes use them without proper understanding [26].

Even though various design guidelines [8, 27, 7] and principles [13] exists, including explanatory prototypes [13, 12], for designing transparent algorithmic UI’s; however, design approaches considering users personality, cognitive style, problem-solving strategies are still unexplored. Research in education, psychology, marketing, and other domains indicate that there exists a significant difference in the way different users use and process information [28, 29, 30]. We do not know how these different cognitive style or mental processes will play out in the case of designing a transparent algorithmic system. Likewise, how much transparency is even enough or desired across different critical audience [31] is also unknown.

We also do not know how to measure a particular user groups transparency needs. To bridge these gaps, we propose a playful exploration approach called “tinkering” [32, 30] as a way of designing transparent algorithmic UI’s by examining Facebook News Feed. There are a couple of benefits of our proposed approach: first, it is possible, that the exploratory nature of algorithm features (matrices) will not overwhelm the user by providing the user sufficient algorithmic control in the personalized interactive news feed experience; second, by enabling the measuring ability at the interface level of how much transparency is desired for each groups, we include the possibility of transparent interface designs that are inclusive (e.g., gender [33, 34].) We do not intend to modify or suggest a new facebook ranking algorithm, instead, our main objective is to encourage a different perspective in the design of transparent algorithmic user interfaces. We end our discussion by suggesting potential future research directions.

## 2. Related Work

### 2.1. Domain Applications and Algorithmic Transparency

Algorithmic systems are everywhere; they range from search engine [35], social media news feed, to video/music or product recommendation systems. These systems has the ability to impact and influence the way we perceive, interact and experience the world around us. Much of the blessings associated with these systems are not free from its perils [36], in many cases, the algorithms are not fair [6]. For example, research showed that Google search algorithm displayed biased and racist content when queried for certain keywords such as “black girls” [35]. The absence of context associated with the search results makes algorithmic interpretation even more difficult [16,

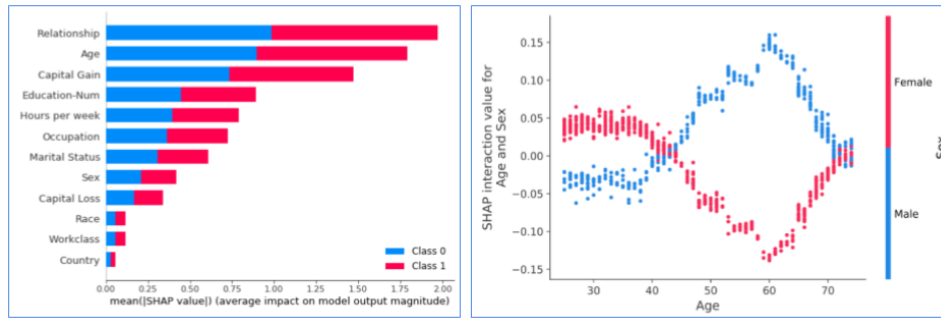
35]. Researchers also discovered biases in image annotation [4] in computer vision across various facets such as race, gender, and weight [3]. As a consequence, researchers advocated for making data's economic value transparent [19], because users will likely stop using a technology due to the lack of opacity about how their generated data is actually used. Algorithmic transparency has also received a considerable amount of attention in data science work practices [37], medical AI applications [38] among many others. The lack of transparency causes mistrust [6] and dissatisfaction in these systems. There exists an increasing opportunity to establish trust through transparency with the advancement of digital media and computer technology [39]. For simplicity, in this paper, we examined algorithmic transparency in the case of Facebook news feed because it is a well-studied socio-technical interactive system [2, 40, 41]. Also, very little is known about how the news feed curation works [41], thus, we wanted to propose an early stage transparent algorithmic news feed prototype, to imagine how a transparent news feed might look like.

## 2.2. Explanation, Interpretability, and Algorithmic Transparency

Previous research have shown the significance of explanation (e.g., how, what, why) to achieve transparency in algorithmic systems such as Facebook news feed curation [19, 2]. Explanation enables the user to become more aware, make judgment about the correctness of the output and mechanism; it also supports interpretability, accountability of algorithmic decision making action [2]. Transparency and interpretability are related through explanation, these relationships are shown in Figure 2. The way existing interpretability tool works is also through explanation [18]. Explana-

tion tools or explainers such as white box and black box API's/packages [18] are available to describe a wide range of ML models. White box explainers (glass box or generalized additive models or GAM's [26]) works directly on data to explain models such as linear regression that is easy to understand; while the black box explainers (post-hoc explanations) requires input and ML models output to explain models that are harder to explain such as neural networks [25]. The explainers are open-source, available in Microsoft's Azure ML packages (e.g., SHAP [42], LIME [32], eli5 [22]), Google Cloud API (e.g., What If Tool [23]) platforms and also in Python Sklearn libraries. Figure 1 shows the output of calling the SHAP summary plot function visualization. These explainers operate on tabular, text, and image data [18, 25].

While these tools and visualizations have helped data scientists to understand the model's output in some cases, but it also depends on the explainers used. Research showed that due to the free availability of these tools, data scientists misuse them by overly trusting them [26, 5]. These tools are mainly used by data scientists and ML practitioners. However, they faced numerous challenges such as model instability (e.g., LIME, SHAP), tools not scaling with large dataset, difficulty in tool integration with their workflow [21]; thus, these tools may not be accessible to technical non-experts (e.g., legal professionals) [26, 21], because applying them requires more than basic programming skill and experience (e.g., knowledge of ML models, built-in methods, see Fig 1). Moreover, we might be able to build transparent data science tools using these explainers, however, we cannot apply them for designing transparent news feed or socio-technical systems (e.g., transparent twitter), because most of these news feed uses proprietary algorithm [40]. Research also showed that explanation may enhance user's positive attitude towards a system, but not necessar-



**Figure 1:** The left image shows, Explainer SHAP [18] depicting feature importance plot, when `shap.summary_plot()` function is called on the model outcome. The right image shows the output of SHAP interaction values, which makes a calls to the function `shap.TreeExplainer(model).shap_interaction_values()` on tree models, to display the interaction among various demographic variables.

ily trust [7]. These limitations encouraged us to discover a distinct way of designing algorithmic transparency at the interface level. Thus, in this study, we propose a cognitive style based approach through incorporating tinkering ability into the design.

### 2.3. Tinkering

Tinkering is a cognitive style or a “mindset” to approach problem-solving through “experimentation and discovery”, [43]; it is associated with exploratory behavior, trial and error method, deviation from instructions when learning [44]. Tinkering is an act of playful experimentation that enhances motivation, influence learning, innovation [45, 44], impact task completion and performance [46, 47, 30]. Even though tinkering is often associated with making activities under playful conditions [48], tinkering behavior has shown to improve learning and educational benefits in domains such as engineering, robots, and programming (e.g., debugging, block-based) [48, 30, 46]. However, tinkering on its own may [44, 48, 45] or may not be a beneficial strategy for problem-solving [30], for example, Beckwith et al. [30]

proposed that effective tinkering happens when it is associated with pause and reflection about software features. We applied tinkering to design transparent algorithmic system with the hope that its exploratory nature will make the overall algorithmic transparency experience less overwhelming. Informed by several existing research [28, 30, 49, 50], the association of gender with respect to tinkering cannot be overlooked, discussed below.

#### 2.3.1. Tinkering and Gender

Previous research has identified gender differences in the tinkering attitude [30, 44, 28], and confirmed that females tend to tinker or explore new software features (e.g., spreadsheet) less compared to the males for problem-solving software [30], also in Computer Science education (e.g., programming assignment) [44]. Numerous studies have showed that tinkering is a mental or psychological trait that distinguishes how different genders (males and females) approach a given task (e.g., making an Arduino project) [44, 47, 28, 30, 48]; tinkering is also one of the facets of gender-inclusive design [49]. Gender-inclusiveness [49] de-

sign do not suggest building a different version of the same software for a different group of user [28], rather, it advocates for designs that support different gender groups equally [51]. Gender inclusivity relies upon five facets of gender differences: motivations, computer self-efficacy, tinkering, information processing style, risk aversion, that can impact the use of problem solving software. While “Inclusive Design considers the full range of human diversity with respect to ability, language, culture, gender, age, and other forms of human difference.” [52], gender is one aspect of inclusive design. Informed and inspired by previous research [51, 34, 30, 28], in this work, we will focus on only gender inclusiveness.

In this study, we discuss tinkering as a means of designing transparent algorithmic UI’s, because of it’s inherent exploratory nature might be less overwhelming [13] to the user while providing personalized user experience; tinkering also adds an additional ability in the design to detect gender differences in the design. Detecting gender issues early on during the design process improves the usability of the software for everyone, including marginalized users [28, 30].

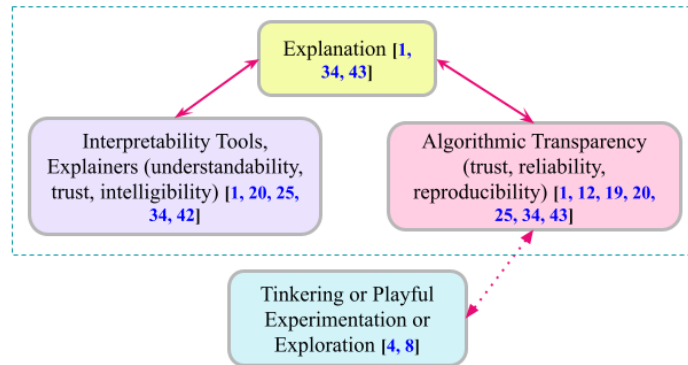
### **3. Designing Transparent Algorithmic User Interface (UI) Through Tinkering**

Transparent algorithmic system for various interactive domain applications will work differently; not only at the user interface level but also at the algorithmic level. For simplicity of design and illustration, we focused on only a single interactive domain, the Facebook news feed. We first provide the rationale behind designing tinkering approach to

transparent UI design, followed by feature description, and a brief discussion of complete transparent news feed prototype, Glass News Feed. Finally, we show how tinkering based approach can be applied to determine how much algorithmic transparency is desired across different user groups, which essentially helps in the determination of gender differences in the interface design.

#### **3.1. Tinkering and Transparent Algorithmic User Interfaces**

Transparent algorithmic UI’s primary objective is to reveal how it works by – explaining the mechanism it uses to produce an outcome [2]. Even though previous research suggests design guidelines for them, however, they did not address the cognitive aspects as a design element in the UI design [8, 27, 13]. We decided to investigate this gap by proposing tinkering based transparent algorithmic UI (see Figure 2). Tinkering is not only related to problem-solving but it is also associated with decision making [28], with regards to software feature (e.g., new or existing) exploration; thus, it allows us with the ability to measure algorithmic transparency needs (how much) across diverse population at the UI level. Following Beckwith et al. [30], we applied the term tinkering as users exploratory behavioral action and practice with software, here news feed features. Allowing the user to “playfully experiment” with the transparent algorithmic features serves two potential purposes: first, the design provides algorithmic information in a manner that is not overwhelming to the user while providing personalized interactive experience; second, tinkering-based design has the ability to be tested for gender inclusiveness issues.



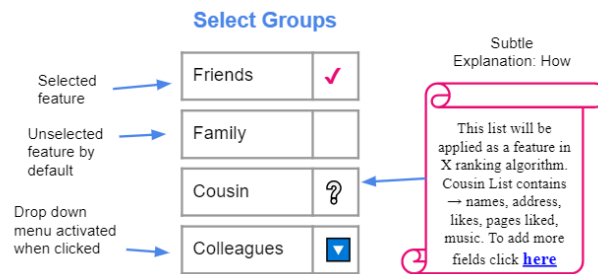
**Figure 2:** The upper portion of the figure illustrates the relationship of Explanation with Transparency and Interpretability. The association of tinkering based approach to algorithmic transparency is depicted below.

### 3.2. Case Study: Facebook Glass News Feed

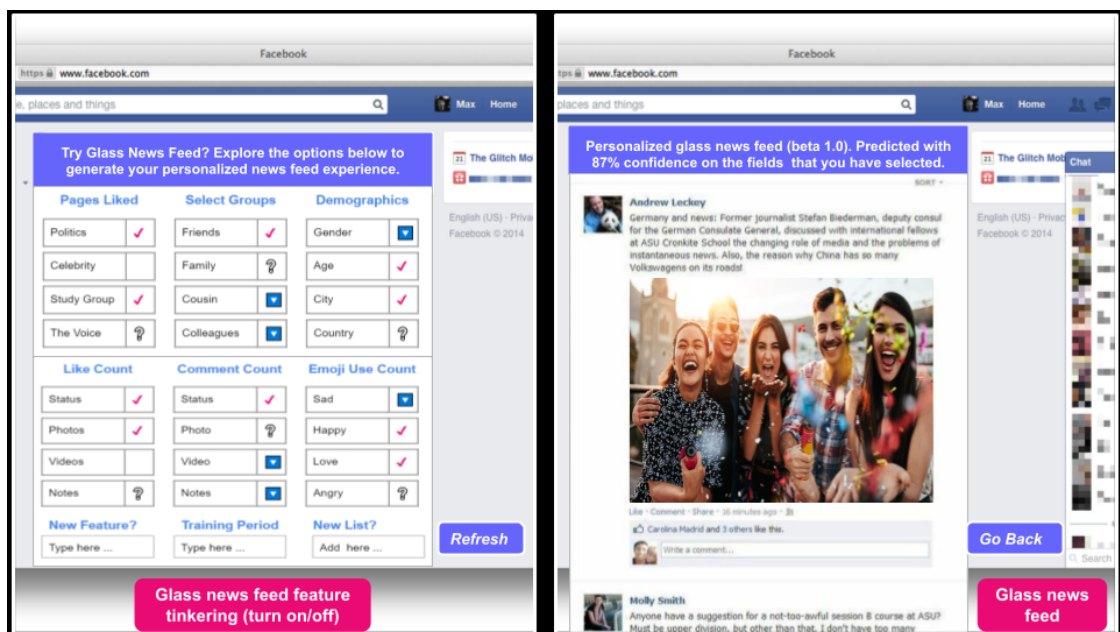
Facebook is one of the most widely used socio-technical systems. Undoubtedly, Facebook has opened numerous opportunities for work, business, collaboration and communication by connecting people worldwide; nonetheless, it has also caused various problems ranging from privacy threats, mental illness, addiction, to users trust violation. Facebook news feed has been well studied in the literature for users perception and understanding of news feed transparency [2, 40, 41]. Facebook news feed works by allowing users to share content and consume content through automated selection and ranking algorithm. The news feed provides content that are relevant, interesting, informative, having high quality [41]. Users are usually unaware of how the underlying algorithmic curation works [2, 53, 41]. Therefore, as a case study, we turned our attention to design transparent algorithmic news feed using a tinkering based approach.

#### 3.2.1. Glass News Feed Features

*Glass News Feed Algorithm Features:* Facebook news feed algorithm applies users past action and behavior data to provide content. Even though existing Facebook news feed provides certain amount of very high-level control about what the user sees and why (e.g., sort, hide, block, follow/unfollow, limited profile) [40], however, a transparent news feed requires other non-trivial control, which revolves around answering “how” question in addition to answering “what” and “why”, but, Facebook news feed blog does not explain that very clearly [41]. For simplicity, we turned users actions in the news feed into transparent algorithmic interface feature (see Figure 3); for example, i) counts such as like and reaction count (e.g., happy, love) in photos, status, videos, ii) list such as friends list, family, acquaintance, and iii) other features such as liked pages (e.g., product, business), public groups user follow, [41], including descriptive features such as notes, tags can used as features. The real future transparent application might apply a different set of features in various categories. We also enabled the ability for the user to be able to create their own feature set and



**Figure 3:** Transparent Glass News Feed feature information represented to support user exploration. For simplicity, a feature can be in any of the following states: selected (✓) indicating on [30], unselected (as empty) indicating off, feature explanation: How (?) option [40, 30]; a click on the drop down menu activates these options for selection.



**Figure 4:** Transparent Glass News Feed prototype applying Facebook user activities such as like counts, groups, emojis count, etc., as features to provide personalized interactive experience (left). When “Refresh” button is clicked, the personalized news feed is displayed (right).

explore the news feed outcome. We showed only some of these features in the proposed prototype.

*Tinkering Capability:* We enabled tinkering

capability in the design through incorporating Facebook data as interface features (see Figure 3); these features can be frequently turned on and off by the user for exploration, as described in [30, 28]. Each of the features

can be in one of the states: i) when checked, marked by ✓, meaning feature selected for current exploration, ii) when un-checked, indicated by an empty box, meaning not currently under exploration, and iii) a question mark (?) to provide feature-related explanation [40]. These capabilities are hidden under the drop down menu, this button is activated when clicked, otherwise, it remains inactive to make sure that these extra abilities does not overwhelm the user. Tinkering count for any particular user can be measured by simply counting the number of features that were turned on and off during a session.

*Subtle Explanation:* Transparency cannot be implemented without providing some kind of explanation. Thus, inspired by Rader et al. [2], we subtly added “How” explanation. “How” news feed with confidence or accuracy information, “Informs participants that the ranking algorithm uses data collected about users and their behaviors to calculate score score for each story”. Explanation “How” was indicated by a question (?) mark, and gets activated when clicked to indicate more information (see Fig 3), by showing other metadata information about the queried feature. This explanation feature becomes really essential when user creates their own “user-defined” feature set for experimentation with the news feed. This ability of defining user-defined feature set ensures enough flexibility for exploration without overwhelming the user with all possible tinkering options.

### 3.2.2. Transparent Glass News Feed

The complete very first prototype of Glass news feed is presented in Figure 4. Tinkering or feature set exploration window is depicted on the left, and the corresponding outcome is shown on the right. For simplicity, we assumed that the features will appear on the news feed itself, though, it is possible to

design the exploration window in many different ways, for different applications.

The design of tinkering enabled transparent algorithmic UI was inspired by the design techniques suggested in problem-solving domain [30]. We added tinkering capabilities in the Glass news feed design for feature set exploration and experiment with corresponding news feed output. The Glass news feed feature sets were derived from relevant research [41], and was kept to a minimum number to avoid causing information overload. We incorporated the ability to add a “user-defined” feature set to provide some flexibility. The entire interactive experience is built on the concept of “playful experimentation” while giving users enough control without hurting their interface experience [13].

The resulting news feed is displayed on the right (top right corner in Figure 4). The algorithmic outcome (news feed after refresh) intentionally provides minimal information such as confidence accuracy, because we have no idea what specific selection or ranking algorithm Facebook originally uses for news feed curation [41]. For similar reason, we did not apply visualization, however, it is a possibility [18, 5, 13]. This is again our very first trying of tinkering approach to achieve algorithmic transparency in interactive systems.

### 3.2.3. Measuring Gender Differences in Glass News Feed

Though, our main motivation for applying tinkering approach to design transparent algorithmic systems was to enable the exploratory nature of tinkering to unfold in the interface design, we suspect that the playful cognitive style might also be able to reduce cognitive load in the transparent systems [20, 15]. Another direct outcome from applying tinkering approach is that it allows the designer to check for gender issues in the design. The



way our proposal is able to detect and measure gender differences is through measuring (counting) “how much” tinkering (on/off) an user engaged during an episode, consistent with prior study of tinkering in problem-solving [30] domain. Similarly, whether our design suffers from gender issues or not can be measured by collecting users tinkering frequency, tinkering episode and tinkering rate. For any particular task (in a user study), i) tinkering frequency is the number of features a user have turned on and off; ii) tinkering episode can be defined as a fixed amount of time for task completion; iii) tinkering rate is the ratio of the previous two measures (i and ii). Depending on the number of user groups taking part in the study, tinkering measures for each user groups can be passed to statistical or ML models for quantitative analysis. We did not show these measures in this study, rather, these are some of the potential areas for future exploration. Transparency is critical for designing interactive social media news feed for trust building and system acceptance. However, too much openness may make the system vulnerable to various kinds of exploitation, harm, and misuse. Thus, how to balance such competing, yet a necessary aspect of a transparent news feed requires further inquiry.

## 4. Limitations and Future Work

In this study, we proposed a tinkering based approach towards designing transparent algorithmic user interface. There are several limitations to this study that is worth mentioning. First, the Glass news feed design was inspired by relevant research in Facebook news feed [41] and tinkering [30, 28]. While background research related to tinkering was broad and detailed, Facebook news feed research was

limited, because Facebook news feed uses proprietary algorithm [40]. A useful workaround suggested in [2] can be beneficial for designing other socio-technical transparent systems, by content analysis of blog posts or related sources. Second, we proposed transparent algorithmic prototype for social media news feed only, there are other algorithmic domain applications such as recommender systems, data science tools, data journalism tools, that can be designed and tested using similar strategy suggested in this study. Our design was also very limited in features and capabilities. Future work might take our design concept, expand (features/matrics) it, and test with the various users to see how tinkering plays out in achieving transparency. Third, we addressed tinkering approach to the design of transparent algorithmic system, however, there are other facets of cognitive styles such as risk-aversion, information processing style (e.g., [49, 34]) which might influence the use of transparent systems (especially for females), we did not address these complex relationships while designing our proposal. Thus, future work should examine other cognitive styles of problem solving and their influence on tinkering when designing transparent algorithmic system. Additionally, most previous studies investigated genders (males and females) influence in design research, thus, we need to expand our understanding by including marginalized LGBTQ+ communities in our design process. Finally, gender is one dimension in the broad spectrum of inclusive design [52], thus, future studies should investigate other diversity dimensions (e.g., race, class, language) while designing transparent systems.

## 5. Conclusion

The demand for transparent algorithmic user interfaces is on the rise. Previous research applied explanations associated with text and

visualization techniques to improve the interpretability of ML models. These specialized tools are mainly used by technical experts such as data scientists and cannot be easily adapted for developing other transparent domain applications such as socio-technical systems. Furthermore, sample transparent UI prototypes in diverse domains exists, however, we do not know how to design a transparent interactive Facebook news feed that do not hurt the UX. Also, how much transparency is even desired across diverse population and how to measure that is also unknown. Thus, in this study, we proposed a very first tinkering based transparent algorithmic Glass News Feed UI prototype with the potential to navigate these multiple scenarios. This proposal can be easily expanded and adapted to design transparent algorithmic systems in various domain applications (e.g., transparent algorithmic tools for the journalists [54]), which essentially requires further examination with various groups of users to understand its technical feasibility, ethical and societal implications (e.g., benefits, harms).

## 6. Acknowledgments

I would like to thank anonymous reviewers for their valuable comments and feedback.

## References

- [1] S. M. Julia Angwin, Jeff Larson, L. Kirchner, Machine bias, 2016. URL: <https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing>, accessed: 2020-07-08.
- [2] E. Rader, K. Cotter, J. Cho, Explanations as mechanisms for supporting algorithmic transparency, in: Proceedings of the 2018 CHI conference on human factors in computing systems, 2018, pp. 1–13.
- [3] K. Wiggers, Researchers show that computer vision algorithms pretrained on imagenet exhibit multiple, distressing biases, 2020. URL: <https://venturebeat.com/2020/11/03/researchers-show-that-computer-vision-algorithms-pretrained-on>, accessed: 2020-04-11.
- [4] M. Miceli, M. Schuessler, T. Yang, Between subjectivity and imposition: Power dynamics in data annotation for computer vision, Proceedings of the ACM on Human-Computer Interaction 4 (2020) 1–25.
- [5] H. Shen, H. Jin, Á. A. Cabrera, A. Perer, H. Zhu, J. I. Hong, Designing alternative representations of confusion matrices to support non-expert public understanding of algorithm performance, Proceedings of the ACM on Human-Computer Interaction 4 (2020) 1–22.
- [6] A. Woodruff, S. E. Fox, S. Rousso-Schindler, J. Warshaw, A qualitative exploration of perceptions of algorithmic fairness, in: Proceedings of the 2018 chi conference on human factors in computing systems, 2018, pp. 1–14.
- [7] R. F. Kizilcec, How much information? effects of transparency on trust in an algorithmic interface, in: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, 2016, pp. 2390–2395.
- [8] M. Eiband, H. Schneider, M. Bilandzic, J. Fazekas-Con, M. Haug, H. Hussmann, Bringing transparency design into practice, in: 23rd international conference on intelligent user interfaces, 2018, pp. 211–223.
- [9] R. Sinha, K. Swearingen, The role of transparency in recommender systems, in: CHI’02 extended abstracts on Human factors in computing systems, 2002, pp. 830–831.

- [10] K. Balog, F. Radlinski, S. Arakelyan, Transparent, scrutable and explainable user models for personalized recommendation, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 265–274.
- [11] M. Kay, S. Haroz, S. Guha, P. Dragicevic, C. Wacharamanotham, Moving transparent statistics forward at chi, in: Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems, 2017, pp. 534–541.
- [12] T. Kulesza, S. Stumpf, M. Burnett, W.-K. Wong, Y. Riche, T. Moore, I. Oberst, A. Shinsel, K. McIntosh, Explanatory debugging: Supporting end-user debugging of machine-learned programs, in: 2010 IEEE Symposium on Visual Languages and Human-Centric Computing, IEEE, 2010, pp. 41–48.
- [13] T. Kulesza, M. Burnett, W.-K. Wong, S. Stumpf, Principles of explanatory debugging to personalize interactive machine learning, in: Proceedings of the 20th international conference on intelligent user interfaces, 2015, pp. 126–137.
- [14] B. Kovach, T. Rosenstiel, The elements of journalism: What newspeople should know and the public should expect, Three Rivers Press (CA), 2014.
- [15] N. Diakopoulos, M. Koliska, Algorithmic transparency in the news media, Digital Journalism 5 (2017) 809–828.
- [16] C. D’Ignazio, L. F. Klein, Data feminism, MIT Press, 2020.
- [17] T. Gebru, J. Morgenstern, B. Vecchione, J. Wortman Vaughan, H. Wallach, H. Daumé III, K. Crawford, Datasheets for datasets, 2018. URL: <https://www.microsoft.com/en-us/research/publication/datasheets-for-datasets/>.
- [18] M. 2020, What is responsible machine learning? (preview), 2020. URL: <https://docs.microsoft.com/en-us/azure/machine-learning/concept-responsible-ml>, accessed: 2020-11-12.
- [19] R. Iyer, Y. Li, H. Li, M. Lewis, R. Sundar, K. Sycara, Transparency and explanation in deep reinforcement learning neural networks, in: Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, 2018, pp. 144–150.
- [20] U. Ehsan, Q. V. Liao, M. Muller, M. O. Riedl, J. D. Weisz, Expanding explainability: Towards social transparency in ai systems, arXiv preprint arXiv:2101.04719 (2021).
- [21] S. R. Hong, J. Hullman, E. Bertini, Human factors in model interpretability: Industry practices, challenges, and needs, Proceedings of the ACM on Human-Computer Interaction 4 (2020) 1–26.
- [22] H. Nori, S. Jenkins, P. Koch, R. Caruana, Interpretml: A unified framework for machine learning interpretability, arXiv preprint arXiv:1909.09223 (2019).
- [23] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viégas, J. Wilson, The what-if tool: Interactive probing of machine learning models, IEEE transactions on visualization and computer graphics 26 (2019) 56–65.
- [24] Z. C. Lipton, The mythos of model interpretability, Queue 16 (2018) 31–57.
- [25] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM computing surveys (CSUR) 51 (2018) 1–42.
- [26] H. Kaur, H. Nori, S. Jenkins, R. Caruana, H. Wallach, J. Wortman Vaughan, 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,

- 2020, pp. 1–14.
- [27] C.-H. Tsai, P. Brusilovsky, Designing explanation interfaces for transparency and beyond., in: *IUI Workshops*, 2019.
- [28] M. Burnett, S. Wiedenbeck, V. Grigoreanu, N. Subrahmanian, L. Beckwith, C. Kissinger, Gender in end-user software engineering, in: *Proceedings of the 4th international workshop on End-user software engineering*, 2008, pp. 21–24.
- [29] J. Meyers-Levy, Gender differences in information processing: A selectivity interpretation, Ph.D. thesis, Northwestern University, 1986.
- [30] L. Beckwith, C. Kissinger, M. Burnett, S. Wiedenbeck, J. Lawrance, A. Blackwell, C. Cook, Tinkering and gender in end-user programmers' debugging, in: *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 2006, pp. 231–240.
- [31] J. Kemper, D. Kolkman, Transparent to whom? no algorithmic accountability without a critical audience, *Information, Communication & Society* 22 (2019) 2081–2096.
- [32] M. T. Ribeiro, S. Singh, C. Guestrin, "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*, 2016, pp. 1135–1144.
- [33] M. M. Burnett, E. F. Churchill, M. J. Lee, Sig: gender-inclusive software: What we know about building it, in: *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, 2015, pp. 857–860.
- [34] S. Stumpf, A. Peters, S. Bardzell, M. Burnett, D. Busse, J. Cauchard, E. Churchill, Gender-inclusive hci research and design: A conceptual review, *Foundations and Trends in Human–Computer Interaction* 13 (2020) 1–69.
- [35] S. U. Noble, *Algorithms of oppression: How search engines reinforce racism*, nyu Press, 2018.
- [36] S. Ballard, K. M. Chappell, K. Kennedy, Judgment call the game: Using value sensitive design and design fiction to surface ethical concerns related to technology, in: *Proceedings of the 2019 on Designing Interactive Systems Conference*, 2019, pp. 421–433.
- [37] M. Muller, M. Feinberg, T. George, S. J. Jackson, B. E. John, M. B. Kery, S. Passi, Human-centered study of data science work practices, in: *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–8.
- [38] B. Haibe-Kains, G. A. Adam, A. Hosny, F. Khodakarami, L. Waldron, B. Wang, C. McIntosh, A. Goldenberg, A. Kundaje, C. S. Greene, et al., Transparency and reproducibility in artificial intelligence, *Nature* 586 (2020) E14–E16.
- [39] A. Meijer, Understanding modern transparency, *International Review of Administrative Sciences* 75 (2009) 255–269.
- [40] E. Rader, R. Gray, Understanding user beliefs about algorithmic curation in the facebook news feed, in: *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, 2015, pp. 173–182.
- [41] K. Cotter, J. Cho, E. Rader, Explaining the news feed algorithm: An analysis of the "news feed fyi" blog, in: *Proceedings of the 2017 CHI conference extended abstracts on human factors in computing systems*, 2017, pp. 1553–1560.
- [42] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: *Advances in neural information processing systems*, 2017, pp.

- 4765–4774.
- [43] D. V. Loertscher, Invent to learn: Making, tinkering, and engineering in the classroom, *Teacher Librarian* 41 (2013) 45.
- [44] S. Krieger, M. Allen, C. Rawn, Are females disinclined to tinker in computer science?, in: *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, 2015, pp. 102–107.
- [45] M. G. Jones, L. Brader-Araje, L. W. Carboni, G. Carter, M. J. Rua, E. Banilower, H. Hatch, Tool time: Gender and students’ use of tools, control, and authority, *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching* 37 (2000) 760–783.
- [46] Y. Dong, S. Marwan, V. Catete, T. Price, T. Barnes, Defining tinkering behavior in open-ended block-based programming assignments, in: *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 2019, pp. 1204–1210.
- [47] M. U. Bers, L. Flannery, E. R. Kazakoff, A. Sullivan, Computational thinking and tinkering: Exploration of an early childhood robotics curriculum, *Computers & Education* 72 (2014) 145–157.
- [48] M. H. Lamers, F. J. Verbeek, P. W. van der Putten, Tinkering in scientific education, in: *International Conference on Advances in Computer Entertainment Technology*, Springer, 2013, pp. 568–571.
- [49] M. Burnett, S. Stumpf, J. Macbeth, S. Makri, L. Beckwith, I. Kwan, A. Peters, W. Jernigan, Gendermag: A method for evaluating software’s gender inclusiveness, *Interacting with Computers* 28 (2016) 760–787.
- [50] D. Showkat, C. Grimm, Identifying gender differences in information processing style, self-efficacy, and tinkering for robot tele-operation, in: *2018 15th International Conference on Ubiquitous Robots (UR)*, IEEE, 2018, pp. 443–448.
- [51] M. Burnett, Doing inclusive design: From gendermag in the trenches to inclusive mag in the research lab, in: *Proceedings of the International Conference on Advanced Visual Interfaces*, 2020, pp. 1–6.
- [52] idrc, Inclusive design research center, 1975. URL: <https://idrc.ocadu.ca/>, accessed: 2021-10-02.
- [53] M. A. DeVito, J. Birnholtz, J. T. Hancock, M. French, S. Liu, How people form folk theories of social media feeds and what it means for how we study self-presentation, in: *Proceedings of the 2018 CHI conference on human factors in computing systems*, 2018, pp. 1–12.
- [54] D. Showkat, E. P. S. Baumer, *Outliers: More than numbers?* (2020).