2021 IEEE 20th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC) | 978-1-6654-2119-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCICC53683.2021.9811333

An Interactive Approach to Bias Mitigation in Machine Learning

Hao Wang[1], Snehasis Mukhopadhyay[1], Yunyu Xiao[2], Shiaofen Fang[1]

1: Department of Computer & Information Science, Indiana University Purdue University Indianapolis
2: Indiana University School of Social Work

Abstract— **Underrepresentation and misrepresentation of protected groups in the training data is a significant source of bias for Machine Learning (ML) algorithms, resulting in decreased confidence and trustworthiness of the generated ML models. Such bias can be mitigated by incorporating both objective as well as subjective (through human users) measures of bias, and compensating for them by means of a suitable selection algorithm over subgroups of training data. In this paper, we propose a methodology of integrating bias detection and mitigation strategies through interactive visualization of machine learning models in selected protected spaces. In this approach, a (partially generated) ML model performance is visualized and evaluated by a human user or a community of human users in terms of potential presence of bias using both objective and subjective criteria. Guided by such human feedback, the ML algorithm can implement a variety of remedial sampling strategies to mitigate the bias using an iterative human-in-the-loop approach. We also provide experimental results with a benchmark ML dataset to demonstrate that such an interactive ML approach holds considerable promise in detecting and mitigating bias in ML models.**

Keywords— fairness, bias, machine learning, visualization, human-computer interaction.

I. Introduction

As machine learning (ML) has been playing an increasingly important role in the decision-making processes in modern society, fairness and bias in ML algorithms have attracted intense research interest in recent years. Although there have been significant progresses made in the detection and mitigation of bias in ML and other AI algorithms, human interactions and community feedback have not been seriously considered part of the solutions to bias detection and mitigation.

"… mathematical formulas alone do not produce consistently relevant results. Human intelligence is still a very important part of the process." [1]. Yet, most of the ML research pays very little, if any, attention to incorporation and utilization of human user participation in the ML process to improve speed, accuracy, and trustworthiness of the results. Broadly, there are two main motivations to include user participation in the ML process. The first is that combining machine algorithms with human intuition and knowledge will help the ML process faster, more efficient, more explainable, and more trusted by human users. In a 1985 seminal paper, Fisher [2] motivated a discussion on optimization/search algorithms that were interactive and allowed humans to be a part of the search process, especially for problems where human thought processes would provide "superior" advantage to the "algorithmic thinking" employed by a computer – for example, processes related to visual perception, strategic thinking, and the ability to learn. According to Fisher's discussions, incorporating human interaction within the optimization algorithms could (a) facilitate model specification and revisions, (b) help cope with problem aspects that are difficult to quantify, and (c) assist in the solution process. The second motivation for interactive machine learning (IML), particularly in socio-technological applications, is that such community participation is the fabric of any democratic society that is a necessary condition for community engagement and stewardship in terms of the generated data-driven solutions. "Democratizing data science", in our view, means ensuring that our discipline promotes the common good, which is often beyond the narrow commercial and demographic interests of the groups that most frequently use data science today" [3].

The need to incorporate human interaction in artificial intelligence (AI) based solutions to complex socio-technological problems, has been recognized by many researchers in multiple domains [4]. Indeed, the recently published 20-year roadmap of AI Research [4] by the Computing Community Consortium (CCC) and Association for the Advancement of Artificial Intelligence (AAAI) identifies "meaningful interaction" as one of the three research priorities, which they define as "comprising techniques for productive collaboration in mixed teams of humans and machines, combining diverse communication modalities (verbal, visual, emotional) while respecting privacy, responsible and trustworthy behaviors that can be corrected directly by users, and fruitful online and real-world interaction among humans and AI systems" [5].

As an automatic method, ML algorithms act mostly as a black box, i.e. the users have very little information about how and why the algorithm work or fail or exhibit bias. Interactive machine learning can provide a mechanism through visualization to allow users to understand and interact with the learning process such that intervention can be properly applied when needed [6]. In the bias detection and mitigation domain, this will allow the users to:

1) Help detect potential bias that may not be obvious from traditional bias detection metrics. Human and community sentiment, intuition and experience play a critical role in the detection implicit biases.

2) Identify the sensitive and non-sensitive variables that are closely related to biased decisions and results so that effective interventions can be applied in a sampling space defined by these variables.

2021 IEEE 20th Int'l Conf. on Cognitive Informatics & Cognitive Computing (ICCI*CC'21)
Y. Wang, J.Z. Wang, H. Leung, N. Howard, P. Soda, B. Widrow & J. Feldman (Eds.)
978-1-6654-2119-5/21/$31.00 ©2021 IEEE

Our goal is to develop a visualization supported interactive ML platform to facilitate user initiated bias detection and mitigation. Through both objective metrics and user feedback, bias in a ML algorithm can be corrected iteratively through progressively reorganizing the training samples (for example, adding new samples with certain conditions). In this paper, we will demonstrate this approach based on a feedback loop with a single user. In practice this process can involve many users and even the community input to take into account of group sentiment.

Interactive Machine Learning is an iterative process in which intermediate ML models are periodically examined by a human user or a community of human users, who provide appropriate performance feedback to the ML algorithm. Based on such feedback, the training dataset is suitably augmented for the next iteration of the ML process. This process of iterative mutual feedback and learning between the ML algorithm and the human user(s) continues until no further improvement is observed. Although this strategy had been previously applied to improve speed of learning and reduce data requirements [7], human factor can potentially play a more important role in bias mitigation because of the subjective nature of bias and farness.

The rest of the paper is organized as follows. In section 2, we review the literatures related to bias detection and mitigation in ML algorithms, as well as some recent work on visualization and interactive techniques in interactive and explainable machine learning. In section 3, we discuss details of the interactive bias mitigation apprach proposed in this paper. In section 4, we describe experimental results with this bias mitigation approach as applied to a benchmark ML dataset. Finally, in section 5, we provide some concluding remarks and directions for further research.

II. Related Work

Fairness and bias in ML have only become a popular research focus in recent years. A large number of research literatures have been published in the past few years addressing a variety of different issues related to fairness and bias in ML algorithms [8, 9]. In this paper, we will primarily focus on ML algorithms for binary classification problem as this is perhaps the most common ML application. Under various formal and informal definitions of AI fairness [10, 11, 12], several different metrics have been proposed to measure bias in ML algorithms [13, 14]. Specifically, metrics for ML-based classification algorithms have been proposed for parity and disparity biases in [15, 16, 17, 18]. Bias metrics related to disparity in prediction accuracies have been discussed in [12, 19].

Based on the fairness definitions and metrics, many techniques have been proposed to mitigate bias by protecting sensitive sociodemographic attributes. Not surprisingly, the class of bias mitigation methods for ML classification problems dominant in the literatures. These bias mitigation methods can generally be classified into Pre-processing, In-Processing and Post-Processing categories [9]. As a pre-processing method, [12] pre-arranges the training set such that the label ratios (e.g. loan approval rates) are the same for groups under a sensitive or protected variable (e.g. racial groups). A related strategy is to omit sensitive variables in the training set [20, 21, 22]. But due to the complex relationships between the sensitive variables and other related variables, these strategies often fail to generate unbiased results and can reduce the model accuracies as well [23]. For this reason, there has been research on analyzing the causal relationship between sensitive and non-sensitive variables [24, 25, 26] to better understand the causes of bias in the dataset so that repairs can be done appropriately in the training set [27]. Unfortunately, causal relationship analysis is not always possible or effective without extensive background information and context.

Another class of methods for mitigating classification bias focus on developing sampling strategies to create training samples that are optimized for fairness, for example, oversampling in areas of decision boundaries [22]. In this approach, the training samples are split into groups based on combinations of values of the sensitive variables [14, 28, 29], and a classifier is trained on each individual subgroup. The selection of the subgroups, which is usually done at the pre-processing stage, can be challenging in some situations. Therefore, some in-processing and post-processing steps, such as recursive partitioning and clustering, may be necessary to prevent issues such as overfitting and other violations of fairness metrics [30, 31].

While interactive and human-in-the-loop techniques have been applied in many areas of AI [32, 33], as far as we know, interactive techniques have not been used for bias mitigation purpose. Due to the complexity of bias and their relationships with sensitive and related no-sensitive variables, human experience, knowledge and perspectives are sometimes more valuable and effective than automatic algorithms in perceiving bias and their potential mitigation solutions. Visualization tools have been used to help users better understand ML and other AI algorithms [34, 35]. Multi-dimensional visualization and graph visualization techniques have previously been applied to depict the relationships between different components of the neural networks [36, 37, 38]. A visual analytics system is proposed in [39] to help ML experts better understand deep convolutional neural networks by clustering the layers and neurons.

Visual analytics methods have been proposed for the performance analysis of ML algorithms in different applications [40, 41]. Interactive methods have also been proposed to improve the performance of ML algorithms through feature selection and optimization during parameter settings [35, 42, 43]. Other performance improvement methods such as training sample selection and model manipulations have also been explored in [44].

The method in [45] applies an incremental training sampling strategy. But it puts a very heavy burden on the user as finding similar images from a large image database or other sources, which can be difficult and time-consuming. A similar strategy is employed in [7], but the goal is to reduce the size of the training sample which is desirable in certain applications.

III. Interactive Bias Mitigation

Interactive approach allows the users to make decisions on what types of training samples are included in order to achieve the desired outcomes. This may include faster speed, data reduction and better accuracy [7]. In the bias mitigation domain, we explore an interactive approach that can identify properties or parameters of training samples that can impact the observed bias in the ML algorithm. This process will iteratively improve the bias problem, caused by inherent data bias, by augmenting/sampling the training data suitably to improve bias metrics of the resulting ML model.

A. *The Iterative Framework*

For a given dataset, let F be the feature space of this dataset, $X \subset F$ be the starting training set, $Y \subset F$ be an internal test set, and *s* be a sensitive variable with a detectable bias over its values $\{s_1, s_2, \dots, s_k\}$ based on the initial ML model. For each learned classification model $M_i(y): F \rightarrow \{0,1\}$ after the i^{th} iteration, we define a visual space V_i, which is formed by a subset of the set of all variables in the original dataset. This subset is identified interactively by the user through visualizing the disparities of the values of each variable with respect to the bias related values of the sensitive variable *s*. Let $Z_i \subset V_i$ be the projection of set Y in the visual space V_i, our tasks in this algorithm are:

(1) Visualize the learned models M_i, as well as their values on the test set Z_i, in the visual space V_i;

(2) Identify areas in V_i in which disparities of misclassified test samples (or other bias metrics) with respect to the values of *s* exist. These are the areas where sampling changes may impact the ML models in reducing the detected bias.

In our experiment, we focus only on adding new training samples $X' \subset F$ in these identified areas such that the learned model $M_{i+1}(y)$ using training set $X \cup X' \subset F$ is an improved model over $M_i(y)$ in terms of the bias metric used. This process continues iteratively until the bias mitigation performance of the model is satisfactory or until the model can no longer be improved. Figure 1 shows a schematic diagram of our interactive and iterative machine learning algorithm for bias mitigation.

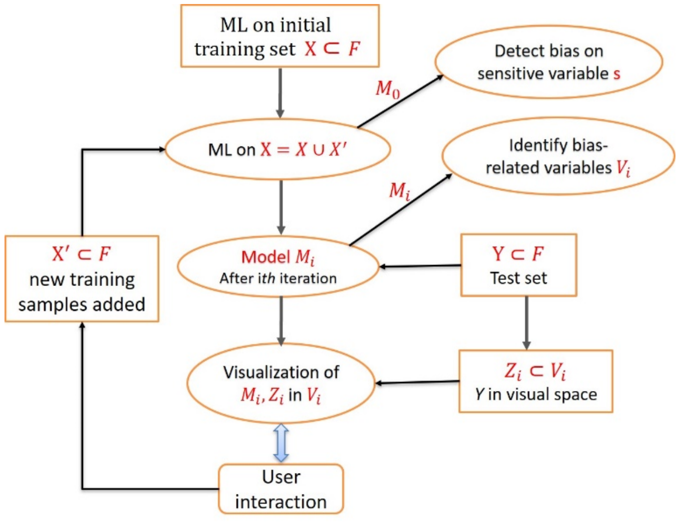
ML on initial training set $X \subset F$
Detect bias on sensitive variable s
M_0
ML on $X = X \cup X'$
Identify bias-related variables V_i
M_i
$X' \subset F$ new training samples added
Model M_i After *ith* iteration
$Y \subset F$ Test set
Visualization of M_i, Z_i in V_i
$Z_i \subset V_i$ *Y* in visual space
User interaction

Figure 1: A schematic flowchart

B. *Bias Detection and Visual Space*

In most applications, biases are detected and measured on a sensitive or protected variable. As discussed in Section 2, many different objective bias detection methods have been proposed to measure bias in a ML model. In a binary classification problem, one common observed bias is the parity and disparity impact in prediction accuracies of the model [12, 19]. For example, a bias exists when the model is consistently more accurate with one protected group over another. Although this type of bias can be automatically detected and measured, interactive bias detection can be valuable when other types of biases are also involved such as community-based fairness issues and counterfactual biases [23]. In our experiment, we use a simple prediction accuracy metric to detect and measure bias, but our approach is interactive and can be applied to most other types of bias metrics as well.

An important strategy in our approach is the separation of feature space and visual space. The visual space is the space in which the ML models and test samples are visualized. It contains variables which are part of the original data attributes, but are likely related to the detected bias. These variables are typically the ones the users can use to select or collect additional samples to add to the training set. Therefore, they need to be easily accessible, i.e. can be obtained from a data sample without significant effort. Since features used in machine learning algorithms are often either pre-computed by some dimension reduction methods (e.g. PCA) or selected through some feature selection algorithms, they are not easily accessible and hence not good variables for the visual space.

To identify the visual space variables, we will first need to examine the behavior of each variable with respect to the different values of the bias variable *s*. If clear disparities can be observed in data distribution patterns, it is an indication that this variable may contribute to the detected bias, and it should be part of the visual space variables. Visualization techniques can be used to help the user observe variable behaviors. In our experiment, we apply a simple box plot technique to view the distributions of training samples on each of the variable axes (Figure 4). Disparities of sample distributions can be easily detected this way.

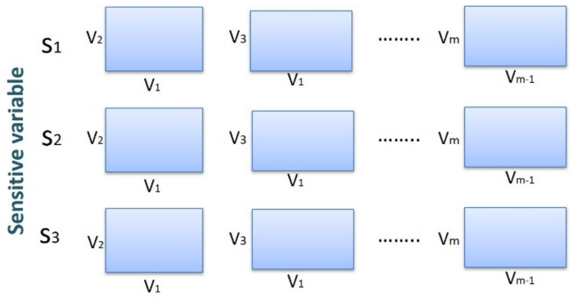

Figure 2: Scatterplot lists

C. *Machine Learning Model Visualization and Interaction*

After each iteration, a new ML model is generated. The ML model is an implicit function defined on the feature space F, but it needs to be visualized in the visual space V_i, along with test results on the test set samples. The visualization method used in this paper is similar to the one in [7]. We use scatterplots to organize the visualization in multiple 2D subspaces of V_i, each representing a cross-section of the model, and a projection of the test samples in this subspace. These scatterplots will be organized sequentially on separate lists. Each list represents a different value of the sensitive variable s for easy comparison to reveal disparities. As shown in Figure 2.

A cumulative projection of a ML model to a 2D subspace does not provide useful information for the user. A carefully designed cross-section (non-linear) is generally more informative. Choosing the proper cross-section is, however, a challenging task. In our visualization design, the ML model will be displayed together with training and testing samples, it is therefore reasonable to generate a cross-section as an interpolation surface $f(x, y)$, defined on the given 2D subspace, which passes through all training samples. The values (of the ML Model) on this cross-section surface will be color-coded within the scatterplot window. This interpolation surface is calculated within a 2D visual space. As described in [7], a triangulation-based interpolation method is applied to capture all pixels within a 2D window. The algorithm interpolates only the feature vectors of the model, which will then be used to compute the model function values for color coding.

Based on the visualization of the ML model and the associated labels of the testing samples, the user can determine where the potential bias problems are in the training set such that adjustment can be made in the training set. There are several possible scenarios or principles that can guide the users' actions:

- **Model Boundary Smoothness**. The boundaries of the ML model can be visually inspected to identify areas with fragmented boundaries, especially if this problem only exists with part of the values of the sensitive variable. This could be an indication that some protected groups under the sensitive variable do not have enough training samples in these areas, leading to different performances for different groups.

- **Testing Errors**. Disparities of testing errors in the test set, such as misclassified samples, provide hints about areas in the visual space where the model performs differently for different protected groups under the sensitive variable. This may mean that additional training samples should be added for some groups in these areas.

- **User Identified Clues.** There may be other types of bias or disparities that the users visually detect from these pairs of scatterplots comparisons. There can also be bias or fairness issues which have been historical or common community concerns that the users may pay additional attention on.

IV. Experimental Results

To test our interactive bias mitigation approach, we apply the algorithm to a real-world benchmark dataset collected by Home Credit, the Home Credit Default Risk dataset [46]. This dataset includes a variety of statistical information from the clients, such as biometric information, credit history, etc. We built a model based on this dataset to predict the clients' repayment abilities, where the predicted result 1 represents that the client has payment difficulties and 0 represents all other cases.

Bias in ML algorithms for loan decisions is a complex issue and have been discussed in literatures [47, 48]. Many types of potential biases can be introduced by ML algorithms for loan decisions due to inherited bias in the training samples from traditional loan decision-making policies and processes. Our discussion in this section is not meant to be a full solution for loan prediction bias problem. Instead, we aim to use this problem as a testbed to illustrate how our interactive approach works to mitigate a perceived bias (which may or may not be a significant problem in real world).

The dataset we use includes 30757 samples, where 7689 samples (about 25% of total samples) were used as test set (or validation set) and 23068 samples (about 75% of total samples) were used as train set in our experiment. Among this train set, we selected 3074 samples (about 1% of total samples) as internal test set to guide the user interaction to improve the accuracy and bias mitigation. The model was trained with a starting training set of 200 samples. 20 samples were added to the training set each time we select a location from the scatterplots to add samples during the iterations.

Some pre-processing was needed before applying the ML algorithm. First, the categorical data was encoded with ordinal encoder to convert the categorical data to ordinal integers. Then all the entries of the data were normalized with the standard scaler to remove the mean and scale to unit variance, and then missing values were filled with the mean value of that feature. A linear SVM model was trained using the starting training set to predict the clients' repayment abilities. The test set was split into different subsets by the values of the protected feature, and the classification prediction accuracies on these subsets were calculated. In our experiment, we chose the clients' gender as a protected (sensitive) variable and the test set was split into two groups - the female samples and the male samples.

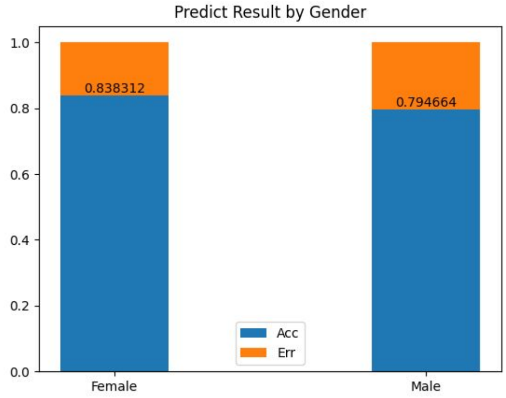

Figure 3: Model accuracies

Figure 3 shows the accuracies of the model on these two groups. Due to the clear difference in prediction accuracy for these two groups, we perceive that a bias exists in this ML classification model against male in terms of prediction accuracy or effectiveness. Our next step is to identify variables that may be related to these two gender groups by visualizing sample distributions with other variables in these two protected groups. Figure 4 shows some examples of the box plots of the data distributions separated by gender. There is some clear data distribution difference in the two groups with the variable "OWN_CAR_AGE" (first row) which indicates that this variable may contribute to the detected bias and can be a good candidate as a visual space variable. Similarly, several other variables were also selected (these variables can be different in different iterations). These include: "FAMILY MEMBERS", "FAMILY STATUS", "LIVE REGION RATING", etc.

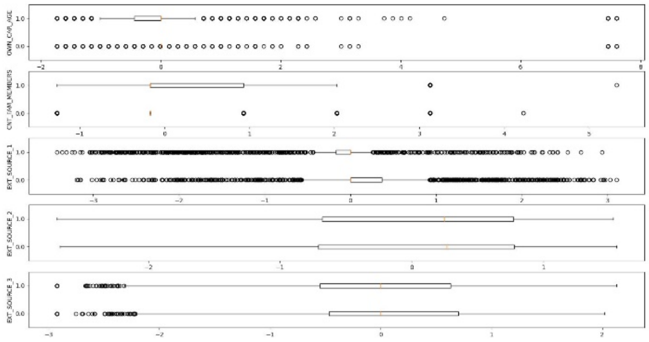

Figure 4: Box plots to show data distributions

After selecting the visual space variables, the scatterplot lists were used to visualize the model and test samples. The user can add new data samples in areas where there is a disparity in misclassified test samples between the two protected groups. Additional samples can also be added in areas with many misclassified samples in both groups (to improve the overall prediction accuracy) and areas the decision boundaries are not well defined. Figure 5 shows the scatter plot lists of one iteration of our experiment. The left panel is for female test samples; the right panel is for male test samples; the background colors represent different predict values of the model; and the triangles represent the misclassified samples on the internal test set. New samples were added in areas where misclassified sample distributions were different between the two groups to help improve the bias in prediction accuracy.

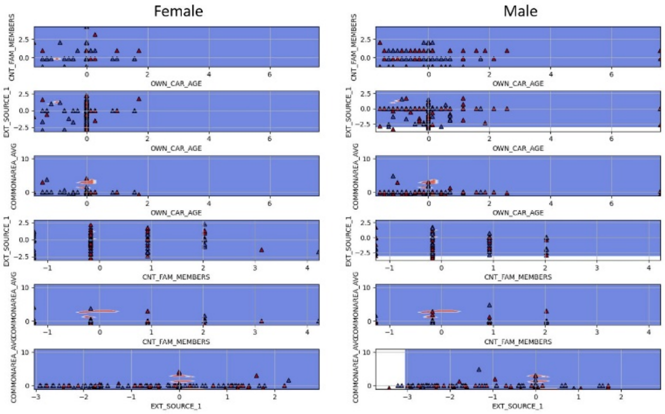

Figure 5: Scatterplot lists for different values of the protected variable

Figure 6 shows the bias mitigation performance in this experiment. We used the difference of the accuracies of the two protected groups to monitor and measure the bias status. The orange line represents the bias mitigation performance by randomly adding the same number of new data samples as that in our algorithm. The blue line represents the bias mitigation performance by interactively selecting new data samples. From Figure 6, we can see that at the beginning the prediction accuracy on the two groups differ by about 6%. By using interactive data selection to mitigate the bias, this difference is reduced quickly. Figure 7 shows the improved accuracies in male group after two iterations. This same effect cannot be achieved by randomly added samples. It is conceivable that this platform will allow the user to explore and experiment other types of actions such as removing samples or adding samples with specific training goals.

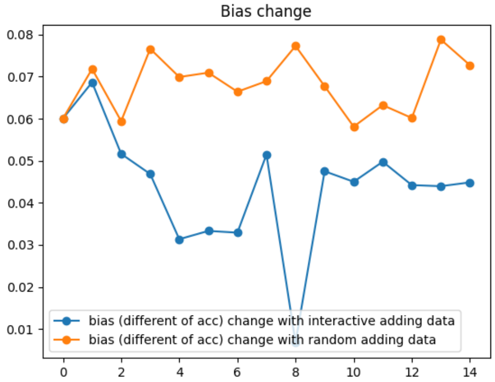

Figure 6: Performance chart for bias mitigation

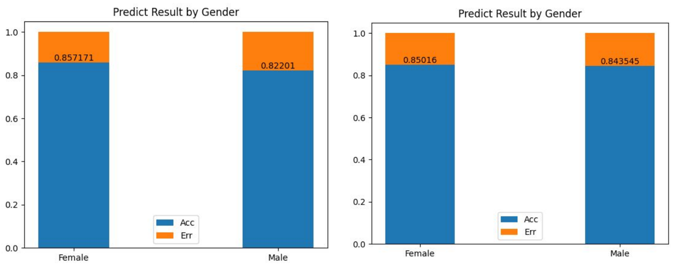

Figure 7: Changes in prediction accuracies after two iterations

V. Conclusions

Bias in the training data or its sampling can degrade the fairness of predictions of ML models, thereby adversely affecting the confidence and trust of the human users. Such fairness degradation can be monitored both by objective fairness criteria, as well as human subjective judgments, utilizing innovative ML model visualization techniques. This paper proposes and develops an interactive machine learning method for iterative bias mitigation by utilizing both visualization, as well as an iterative sampling strategy for training data provided to the ML algorithm. We demonstrate, via experimental studies with a benchmark dataset, that such an interactive approach has the potential to mitigate ML bias by integrating both human intuition and judgment as well as objective measures of bias.

While this paper introduces such an interactive approach to bias mitigation, several research questions remain open that can be further investigated within this framework. The modalities of interactions between the human user and the ML algorithm, particularly for technically challenged users, can be broadened beyond only visualization of ML models. Also, the iterative sampling methodology needs to be adapted to problems where multiple potentially mutually conflicting criteria for fairness and bias exist, as for example, representing multiple users in a user community. This naturally fits into the well-known paradigm of multi-criteria decision-making that are prevalent in any democratic, social decision problem.

REFERENCES

[1] Wales, J., Founder of Wikipedia. Source - Businessweek.com, 12/2006.

[2] Fisher, M.L., 1985. Interactive optimization. Annals of Operations Research, 5(3), pp.539-556.

[3] Chou, S., Li, W. and Sridharan, R., 2014, August. Democratizing data science. In Proceedings of the KDD 2014 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA (pp. 24-27).

[4] Michelucci, P. and Dickinson, J.L., 2016. The power of crowds. Science, 351(6268), pp.32-33.

[5] Gil, Y. and Selman, B., 2019. A 20-year community roadmap for artificial intelligence research in the US. arXiv preprint arXiv:1908.02624.

[6] Amershi, S., Cakmak, M., Knox, W.B. and Kulesza, T., 2014. Power to the people: The role of humans in interactive machine learning. AI Magazine, 35(4), pp.105-120.

[7] Li, H., Fang, S., Mukhopadhyay, S., Saykin, A.J. and Shen, L., 2018, December. Interactive Machine Learning by Visualization: A Small Data Solution. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 3513-3521). IEEE.

[8] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. arXiv preprint arXiv:1908.09635, 2019.

[9] S Caton, C Haas. Fairness in Machine Learning: A Survey. arXiv preprint arXiv:2010.04053, 2020

[10] Harini Suresh and John V Guttag. A framework for understanding unintended consequences of machine learning. arXiv preprint arXiv:1901.10002, 2019.

[11] Silvia Chiappa. Path-specific counterfactual fairness. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 7801–7808, 2019.

[12] Moritz Hardt, Eric Price, Nati Srebro, and Others. Equality of opportunity in supervised learning. In Advances in neural information processing systems, pages 3315–3323, 2016.

[13] Solon Barocas, Moritz Hardt, and Arvind Narayanan. Fairness and Machine Learning. fairmlbook.org, 2019

[14] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In Proceedings of the 26th International Conference on World Wide Web, pages 1171–1180, 2017.

[15] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 259–268, New York, New York, USA, 2015. ACM, ACM Press. ISBN 9781450336642.

[16] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 797–806, New York, New York, USA, 2017. ACM, ACM Press.

[17] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. Fairness in criminal justice risk assessments: The state of the art. Sociological Methods & Research, page 0049124118782533, 2018.

[18] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Allison Woodruff, Christine Luu, Pierre Kreitmann, Jonathan Bischof, and Ed H Chi. Putting fairness principles into practice: Challenges, metrics, and improvements. In Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, pages 453–459, 2019

[19] Till Speicher, Hoda Heidari, Nina Grgic-Hlaca, Krishna P. Gummadi, Adish Singla, AdrianWeller, and Muhammad Bilal Zafar. A Unified Approach to Quantifying Algorithmic Unfairness. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD '18, pages 2239–2248, 2018. ISBN 9781450355520.

[20] Sara Hajian and Josep Domingo-Ferrer. A methodology for direct and indirect discrimination prevention in data mining. IEEE transactions on knowledge and data engineering, 25(7):1445–1459, 2012.

[21] Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R. Varshney. Optimized Pre-Processing for Discrimination Prevention. In Advances in Neural Information Processing Systems, pages 3992–4001, 2017.

[22] Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. Knowledge and Information Systems, 33(1):1–33, oct 2012. ISSN 0219-1377

[23] Hao Wang, Berk Ustun, and Flavio Calmon. Repairing without retraining: Avoiding disparate impact with counterfactual distributions. In International Conference on Machine Learning, pages 6618–6627, 2019

[24] Silvia Chiappa and William S Isaac. A causal bayesian networks viewpoint on fairness. In IFIP International Summer School on Privacy and Identity Management, pages 3–20. Springer, 2018

[25] Bruce Glymour and Jonathan Herington. Measuring the biases that matter: The ethical and casual foundations for measures of fairness in algorithms. In Proceedings of the Conference on Fairness, Accountability, and Transparency, pages 269–278, 2019.

[26] Razieh Nabi and Ilya Shpitser. Fair inference on outcomes. In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[27] Babak Salimi, Luke Rodriguez, Bill Howe, and Dan Suciu. Interventional fairness: Causal database repair for algorithmic fairness. In Proceedings of the 2019 International Conference on Management of Data, pages 793–810, 2019.

[28] Cynthia Dwork, Nicole Immorlica, Adam Tauman Kalai, and Max Leiserson. Decoupled classifiers for groupfair and efficient machine learning. In Conference on Fairness, Accountability and Transparency, pages 119–133, 2018.

[29] Berk Ustun, Yang Liu, and David Parkes. Fairness without harm: Decoupled classifiers with preference guarantees. In International Conference on Machine Learning, pages 6373–6382, 2019.

[30] Vasileios Iosifidis, Besnik Fetahu, and Eirini Ntoutsi. Fae: A fairness-aware ensemble framework. arXiv preprint arXiv:2002.00695, 2020.

[31] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei StevenWu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In International Conference on Machine Learning, pages 2564–2572, 2018.

[32] Amershi, S., Cakmak, M., Knox, W.B. and Kulesza, T., 2014. Power to the people: The role of humans in interactive machine learning. AI Magazine, 35(4), pp.105-120.

[33] Ware, M., Frank, E., Holmes, G., Hall, M. and Witten, I.H., 2001. Interactive machine learning: letting users build classifiers. International Journal of Human-Computer Studies, 55(3), pp.281-292.

[34] Jing Xia, Wei Chen, Yumeng Hou, Wanqi Hu, Xinxin Huang, David S. Ebert. DimScanner: A Relation-based Visual Exploration Approach Towards Data Dimension Inspection. VAST 2016.

[35] Shixia Liu, Xiting Wang, Mengchen Liu, Jun Zhu. Towards better analysis of machine learning models: A visual analytics perspective. Visual Informatics 1 (2017) 48–56.

[36] Zahavy, T., Ben-Zrihem, N., Mannor, S. 2016. Graying the black box: Understanding dqns. In: ICML pp. 1899–1908.

[37] Rauber, P.E., Fadel, S., Falcao, A., Telea, A., 2017. Visualizing the hidden activity of artificial neural networks. IEEE TVCG 23 (1), 101–110.

[38] Tzeng, F.Y., Ma, K.L. 2005. Opening the black box - data driven visualization of neural networks. In: IEEE Visualization, pp. 383–390. http://dx.doi.org/10.1109/VISUAL.2005.1532820.

[39] Liu, M., Shi, J., Li, Z., Li, C., Zhu, J.J.H., Liu, S., 2017. Towards better analysis of deep convolutional neural networks. IEEE TVCG 23 (1), 91–100. http://dx.doi.org/10.

[40] Ren, D., Amershi, S., Lee, B., Suh, J., Williams, J.D., 2017. Squares: Supporting interactive performance analysis for multiclass classifiers. IEEE TVCG 23 (1), 61–70.

[41] Alsallakh, B., Hanbury, A., Hauser, H., Miksch, S., Rauber, A., 2014. Visual methods for analyzing probabilistic classification data. IEEE TVCG 20 (12), 1703–1712.

[42] Krause, Josua, Perer, Adam, and Ng, Kenney. Interacting with predictions: Visual inspection of black-box machine learning models. ACM CHI 2016, 2016.

[43] Wang, J; Fang, S; Li, H; Goni, J; Saykin, AJ; Shen, L. Multigraph Visualization for Feature Classification of Brain Network Data. EuroVis Workshop on Visual Analytics (EuroVA), pp.61-65, 2016.

[44] Liu, M., Liu, S., Zhu, X., Liao, Q., Wei, F., Pan, S., 2016. An uncertainty-aware approach for exploratory microblog retrieval. IEEE TVCG 22 (1), 250–259.

[45] Paiva, J.G.S., Schwartz, W.R., Pedrini, H., Minghim, R., 2015. An approach to supporting incremental visual data classification. IEEE TVCG 21 (1), 4–17.

[46] Home Credit. Home Credit Default Risk. https://www.kaggle.com/c/home-credit-default-risk/data. 2018

[47] Featherstone, Allen M., et al. Factors affecting the agricultural loan decision-making process. No. 1290-2016-102299. 2005.

[48] Bartholomae, Suzanne, et al. "Framing the human capital investment decision: Examining gender bias in student loan borrowing." Journal of Family and Economic Issues 40.1 (2019): 132-145.