Responsible AI Challenges in End-to-end Machine Learning

Steven Euijong Whang, Ki Hyun Tae, Yuji Roh, Geon Heo
KAIST
{swhang, kihyun.tae, yuji.roh, geon.heo}@kaist.ac.kr

Abstract

Responsible AI is becoming critical as AI is widely used in our everyday lives. Many companies that deploy AI publicly state that when training a model, we not only need to improve its accuracy, but also need to guarantee that the model does not discriminate against users (fairness), is resilient to noisy or poisoned data (robustness), is explainable, and more. In addition, these objectives are not only relevant to model training, but to all steps of end-to-end machine learning, which include data collection, data cleaning and validation, model training, model evaluation, and model management and serving. Finally, responsible AI is conceptually challenging, and supporting all the objectives must be as easy as possible. We thus propose three key research directions towards this vision – depth*,* breadth*, and* usability *– to measure progress and introduce our ongoing research. First, responsible AI must be deeply supported where multiple objectives like fairness and robust must be handled together. To this end, we propose FR-Train, a holistic framework for fair and robust model training in the presence of data bias and poisoning. Second, responsible AI must be broadly supported, preferably in all steps of machine learning. Currently we focus on the data pre-processing steps and propose Slice Tuner, a selective data acquisition framework for training fair and accurate models, and MLClean, a data cleaning framework that also improves fairness and robustness. Finally, responsible AI must be usable where the techniques must be easy to deploy and actionable. We propose FairBatch, a batch selection approach for fairness that is effective and simple to use, and Slice Finder, a model evaluation tool that automatically finds problematic slices. We believe we scratched the surface of responsible AI for end-to-end machine learning and suggest research challenges moving forward.*

1 Introduction

Responsible AI is becoming critical as machine learning becomes widespread in our everyday lives. Companies including Google [2], Microsoft [3], and IBM [5] publicly state that AI not only needs to be accurate, but also used and developed, evaluated, and monitored for trust. Although there is no universally agreed notion for responsible AI, the major objectives include fairness, robustness, explainability, transparency, and accountability.

The usual starting point is to support responsible AI only in model training, but this is not sufficient. For example, if the training data is biased towards a specific population, there is a fundamental limit into how much the trained model can avoid being biased as well even using the best fair training algorithms. Instead, we may need to address the root cause starting from data collection where we need to construct an unbiased dataset.

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

We would thus like to support responsible AI in all steps of end-to-end machine learning [8, 37]. Before model training, the key steps are data collection, data cleaning, and validation. After model training, there are model evaluation, and model management and serving. In addition, since supporting all the responsible AI objectives is already conceptually challenging, it is important to make these techniques easy to use as well.

To this end, we propose three research directions – *depth*, *breadth*, and *usability* – and present our contributions. First, we need to deeply support responsible AI where multiple objectives are addressed together. We present FR-Train [28], the first holistic framework for fair and robust model training. Second, we need to broadly support responsible AI in all machine learning steps. We present two systems that focus on data pre-processing: Slice Tuner [34] is a selective data acquisition framework for fair and accurate models, and MLClean [33] is a data cleaning framework that also improves fairness and robustness. Third, we need responsible AI to be usable and actionable. We present two systems: FairBatch [29] is an easy-to-deploy batch selection technique for model training that improves fairness, and Slice Finder [13, 14] automatically evaluates a model by finding problematic slices where it underperforms. Our work only scratches the surface of responsible AI for end-to-end machine learning, and we believe that setting the three research directions is useful to measure progress.

We introduce the responsible AI research landscape in Section 2. We then discuss our systems for depth, breadth, and usability in Sections 3, 4, and 5, respectively. Finally, we suggest open challenges in Section 6.

2 Responsible AI Research Landscape

We provide a brief history of responsible AI and discuss the research landscape. Responsible AI is also known as Trustworthy AI and has recently been promoted by Google [2], Microsoft [3], and IBM [5] among others as a critical issue when using AI in practice. The key objectives include fairness, robustness, explainability, transparency, and accountability. Among the objectives, we focus on fairness and robustness because they are both closely related to the training data. The other objectives are also important, but currently outside our scope.

Fairness is the problem of not discriminating against users and has gained explosive interest in the past decade [7, 35]. An article that popularized fairness was the 2016 ProPublica report [6] on the COMPAS software, which is used in US courts to predict a defendant's recidivism (reoffending) rate. COMPAS is convenient, but is known to overestimate black people's recidivism risk compared to white people. Recently, various unfairness mitigation techniques [9] have been proposed and can be categorized as pre-processing, in-processing, or post-processing depending on whether the techniques are applied before, during, or after model training, respectively.

Robustness is the problem of preventing or coping with adversarial attacks. In particular, model training against data poisoning has been heavily studied in the past decade [15, 31]. Nowadays datasets are easier to publish using tools like Kaggle and Google Dataset Search [11], which means that it is easier to disseminate poisoned data as well. The data can then be harvested by Web crawlers of unsuspecting victims and used for model training. While the basic poisoning attacks involve simple labeling flipping (e.g., change a positive label to be negative), recent poisoning attacks are becoming increasingly sophisticated. The possible defenses include sanitizing the data before model training or making the model training accurate despite the poisoning.

In practice, machine learning is not just about model training, but involves multiple steps as demonstrated by end-to-end systems like TensorFlow Extended (TFX) [8] and MLFlow [37]: data collection, data cleaning and validation, model training, model evaluation, and model management and serving. Hence, responsible AI is not just a model training issue, but relevant to all of the above steps. The data management community has recently been addressing the data aspect of responsible AI in end-to-end machine learning [25, 23, 26, 12, 27, 32, 36].

The current research landscape naturally leads to the three key research directions we propose – *depth*, *breadth*, and *usability* – as shown in Figure 1. First, it is important to support many responsible AI objectives at each step. Second, we need to broadly support responsible AI in as many steps as possible, from data collection to model serving. Third, we need these techniques to be usable and actionable by machine learning users. We highlight the responsible AI objectives in Figure 1 where we propose solutions.

3 Deep Responsible AI

We discuss deeply supporting responsible AI, which means that we would like to address multiple objectives together. We re-emphasize that each objective is currently being heavily studied. For model fairness, there is an extensive literature in the machine learning and fairness communities on mitigating unfairness before, during, or after model training [7, 35, 9]. For model robustness, both the machine learning and security communities are proposing various data sanitization and robust training techniques [22, 31]. However, we believe that responsible AI requires both fairness and robustness instead of just one. In addition, addressing one objective at a time is not ideal as we discuss later. Fairness and robustness are also closely related because their problems originate from the training data: biased data causes unfairness while poisoned data decreases model accuracy. This motivation leads us to propose FR-Train [28], the first holistic framework for fair and robust training.

Fairness is a subjective notion, and many definitions have been proposed [35] where they can be categorized depending on what information is used: the classifier, the sensitive attribute (e.g., race or gender), and training labels. For example, individual fairness only uses the classifier and means that similar individuals must have similar predictions. Demographic parity [17] (or disparate impact) uses the classifier and the protected attribute and means that different sensitive groups (e.g., black and white populations) have similar positive prediction rates. That is, $P(\hat{Y} = 1|Z = 0) \approx P(\hat{Y} = 1|Z = 1)$ where $\hat{Y}$ is a prediction and Z is a binary sensitive attribute. Equalized odds [18] uses all three pieces of information and is similar to demographic parity, except that the probabilities are conditioned on the label. That is, $P(\hat{Y} = 1|Z = 0, Y = l) \approx P(\hat{Y} = 1|Z = 1, Y = l)$ where Y is the label. In this section, we use demographic parity and measure it using the formula $DP := \min\left(\frac{P(\hat{Y}=1|Z=0)}{P(\hat{Y}=1|Z=1)}, \frac{P(\hat{Y}=1|Z=1)}{P(\hat{Y}=1|Z=0)}\right)$ where a higher value close to 1 means better fairness.

We now explain why addressing fairness and robustness together is important using a concrete example. In Figure 2, suppose there are two sensitive groups black and white, and that there are ten people of two races: white (denoted as 'w') and black (denoted as 'b'). Let us assume the boxes indicates positive labels and that we want to train a threshold classifier that divides the individuals using a single feature X where those on the left have negative predictions (e.g., do not reoffend) while those on the right have positive predictions. On clean data, a vanilla classifier can obtain perfect accuracy by dividing between the fourth and fifth individuals (Figure 2 solid line classifier). However, the demographic parity DP is not perfect where $P(\hat{Y} = 1|Z = w) = \frac{2}{5} = 0.4$, $P(\hat{Y} = 1|Z = b) = \frac{4}{5} = 0.8$, and $DP := \min\left(\frac{0.4}{0.8}, \frac{0.8}{0.4}\right) = 0.5$. Suppose a fair classifier maximizes accuracy with perfect DP. One can find such a classifier by dividing between the second and third individuals (Figure 2 blue dotted line classifier). While $DP = 1$, the accuracy is 0.8 because two white people are now misclassified.

Now suppose we poison the clean data by using the standard method of flipping labels [24]. On the bottom of the left side of Figure 2, the fifth and seventh individuals are now incorrectly labeled as negative. There are three ways to handle the poisoned data: (1) do nothing and perform fair training only as usual, (2) take a two-step approach and perform data sanitization followed by fair training using existing methods, and (3) take a holistic approach for fair and robust training. Let us first see what happens if we take the first approach. We can train a fair classifier on the poisoned data with perfect DP by dividing between the eighth and ninth individuals (bottom of the right side of Figure 2, red dotted line classifier). In that case, we will have perfect DP, but an accuracy of 0.8 on poisoned data. However, if this classifier is deployed in the real world, it will effectively be used on clean data. This scenario is plausible for any application that serves real customers. However, simply using the same classifier on clean data results in a worse tradeoff of fairness and accuracy where DP remains the same, but the accuracy reduces to 0.6. Hence, ignoring poisoning may lead to strictly worse accuracy and fairness results. In reference [28], we also empirically show that the two-step solution is ineffective. The intuition is that an existing fairness-only or robustness-only technique cannot easily distinguish data poisoning from bias in the data and ends up removing all or none of the problematic data.

We thus propose FR-Train to take a holistic approach for fair and robust training. Figure 3 shows the architecture of FR-Train. On the top, there is a classifier (e.g., predicts recidivism) that competes with a discriminator

for fairness that predicts the sensitive attribute (e.g., the race) based on the predictions. This adversarial training is similar to Adversarial Debiasing [38], a state-of-the-art fairness-only training algorithm. The below part is the novel addition where there is a discriminator for robustness that distinguishes the possibly-poisoned training set with a validation set that is known to be clean. The clean validation set is small and can be constructed using crowdsourcing and conventional quality control techniques including majority voting. Hence, the classifier needs to be both fair and robust to compete with the two discriminators. Finally, the predictions of the robustness discriminator are used to reweight training set examples where cleaner examples get higher weights. Initially, these weights are not useful because the robustness discriminator is not accurate. However, as the training progresses, the discriminator becomes accurate, and the weights are used by the classifier.

In reference [28], we present a mutual information-based interpretation of FR-Train's architecture. To give an intuition, perfect fairness means that the mutual information between the model's prediction and the sensitive attribute is 0. Similarly, satisfying robustness can be expressed using mutual information. In this case, perfect robustness means that the poisoned data distribution is indistinguishable from the clean data distribution (i.e., validation set). FR-Train minimizes both of the mutual information values and the classifier loss. We perform experiments on synthetic and real datasets and train a classifier on poisoned data and evaluate it on clean data. As a result, FR-Train is the only approach that achieves both high accuracy and fairness while the other baselines either have poor fairness or accuracy.

4 Broad Responsible AI

In addition to supporting responsible AI in model training, we would also like to broadly support it across many steps in end-to-end machine learning. While most of the fairness and robustness literature focus on model training, there needs to be more focus on other machine learning steps as well. Recently, FairPrep [30] was proposed to support fairness in all steps of data pre-processing before model training. Also for an extensive coverage of data collection and quality techniques for machine learning, please refer to a survey [27] and tutorial [36]. Here we also focus on data pre-processing and present two contributions: Slice Tuner [34] is a selective data acquisition framework for maximizing fairness and accuracy, and MLClean [33] is a data cleaning framework for addressing both fairness and robustness in addition to accuracy.

4.1 Selective Data Acquisition for Fair and Accurate Models

As machine learning is used in various applications, one of the critical bottlenecks is acquiring enough data so that the trained model is both accurate and fair. Nowadays, there are many ways to acquire data including dataset discovery, crowdsourcing, and simulator-based data generation. Data acquisition is not the same as active learning, which labels existing data. Instead, our focus is on acquiring new data along with its labels.

However, blindly acquiring data is not the right approach. Let us first divide the data into subsets called slices. Suppose that the slices are customer purchases by various regions: America, Europe, APAC, and so on. Among them, if we already have enough America data, acquiring more America data is not only unhelpful, but may also bias the data and have a negative effect on the model accuracy on the other slices.

Instead, we want to acquire possibly different amounts of data per slice in order to maximize accuracy and fairness. To measure accuracy, we use loss functions like logistic loss. For fairness, we use equalized error rates [35], which states that the losses of slices must be similar. This notion of fairness is important to any application that should not discriminate its customers by service quality. A waterfilling approach is a good start where we simply acquire data so that the slices have similar sizes. However, this approach is not optimal because some slices may need more data to obtain the same model loss as other slices.

Our key approach is to generate for each slice a *learning curve*, which estimates the model loss on that slice given more labeled data. Multiple studies [19, 16] show that a learning curve is best fit using a power-law

function. Figure 4 (a) shows two actual learning curves generated on two race-gender slices of a real dataset called UTKFace [39]. We can use these learning curves to estimate how much data must be acquired per slice.

Assuming that the learning curves are perfectly reliable (we discuss how to deal with unreliable curves later), we can determine the amounts of data to acquire to minimize the total loss and unfairness of slices by solving the following convex optimization problem:

$$\min \sum_{i=1}^{n} b_i(|s_i| + d_i)^{-a_i} + \lambda \sum_{i=1}^{n} \max\left\{0, \frac{b_i(|s_i| + d_i)^{-a_i}}{A} - 1\right\} \text{ subject to } \sum_{i=1}^{n} C(s_i) \times d_i = B$$

where $\{s_i\}_{i=1}^n$ are the slices, $\{d_i\}_{i=1}^n$ are the amounts of data to acquire, A is the average loss of slices, $C(s_i)$ is the cost function for acquiring an example for s_i, and B is a cost budget. The first term in the objective function minimizes the total loss while the second term minimizes the unfairness by penalizing slices that have higher-than-average losses. The two terms are balanced using λ. By acquiring more data for slices with higher losses, we eventually satisfy equalized error rates. Slice Tuner's architecture is shown in Figure 4 (b) where we perform selective data acquisition on input slices. The runtime bottleneck is the time to actually acquire data.

We now address the key challenge of handling unreliable learning curves. Learning curves are not perfect because slices may be too small for accurate estimations. Even worse, acquiring data for one slice may "influence" others. Figure 5 (a) shows how acquiring data for the slice White-Male increases or even decreases the model's loss on other slices for UTKFace. The intuition is that the acquired data of one slice pushes the decision boundary of the model, which in turn changes the losses of other slices (Figure 5 (b)).

The solution is to iteratively update the learning curves. But how often should we iterate? On one hand, each iteration is expensive and involves multiple model trainings and curve fittings, even though we use amortization techniques [34]. On the other hand, we do not want to use inaccurate learning curves. Our algorithm works as follows. We first ensure a minimum slice size to draw some learning curve. In practice, having tens of examples is enough for this step. Next, we repeat two steps until we run out of budget: (1) acquire data as long as the estimated influence is not large enough and (2) re-fit the learning curves. The remaining problem is estimating influence. We propose a proxy called imbalance ratio change where imbalance ratio represents bias and is the ratio between the largest and smallest slice sizes. The intuition is that a change in imbalance ratio among slices causes influence. In Figure 5 (b) adding two triangles results in a shifted decision boundary where the imbalance ratio increases from $\frac{2}{2} = 1$ to $\frac{4}{2} = 2$. On the other hand, if we evenly increase the slices, the decision boundary does not shift, and the imbalance ratio does not change much either.

In reference [34], we provide more details on the algorithms and also perform experiments on real datasets. We show that Slice Tuner has lower loss and unfairness compared to two baselines: uniformly acquiring the same amounts of data per slice and waterfilling. We also make the same comparison when the slices are small and only have tens of examples. Here the learning curves are very noisy and thus unreliable. Interestingly, Slice Tuner still outperforms the baselines because it can still leverage the relative loss differences among the learning curves. As more data is acquired, Slice Tuner performs even better with more reliable learning curves. In the worst case when the learning curves are completely random, we expect Slice Tuner to perform similarly to one of the baselines.

4.2 Data Cleaning for Accurate, Fair, and Robust Models

Another important place to support responsible AI is data cleaning [20] where the input data needs to be validated and fixed before it is used for model training. Historically, multiple communities – data management, machine learning (model fairness), and security – have been investigating this problem under the names of data cleaning, unfairness mitigation, and data sanitization, respectively. Unfortunately, not much is known how the different techniques can be used together when a dataset is dirty, biased, and poisoned at the same time.

MLClean is a unified cleaning framework that performs data cleaning, data sanitization, and unfairness mitigation together. A key insight is that these three operations have dependencies and must be executed in

ID	Weight	Name	Gender	Age	Label
e_1	1.0	John	M	20	1
e_2	1.0	Joe	M	20	0
e_3	1.0	Joseph	M	20	0
e_4	1.0	Sally	F	30	1
e_5	1.0	Sally	F	40	0
e_6	1.0	Sally	F	300	1

Table 1: Six examples where e_2 and e_3 are duplicates (dirty), and e_6 has an anomalous age (poisoned).

a certain order for the best performance. As shown in MLClean's architecture in Figure 7, data sanitization and cleaning are performed together followed by unfairness mitigation. Data sanitization can be considered a stronger version of cleaning because it defends against adversarial poisoning instead of just noise. In addition, data cleaning and sanitization may affect the bias of data while unfairness mitigation that performs example reweighting does not affect the correctness of cleaning and sanitization.

As a running example, suppose we run MLClean on the examples in Table 4.2 with equal weights of 1. Say that data sanitization clusters examples and removes anomalies while data cleaning performs entity resolution. The two operations can be naturally combined by generating clusters and running entity resolution within each cluster, assuming that examples across clusters do not match. Clustering examples before resolution is a common operation in entity resolution for narrowing down matching candidates. Figure 7 shows how the initial six examples are clustered into $\{e_1, e_2, e_3\}$ and $\{e_4, e_5\}$ (e_6 is considered an outlier), and then e_2 and e_3 are merged together into e_{23} with a summed weight of 2. For unfairness mitigation, suppose we reweight [21] the examples such that demographic parity (defined in Section 3) is satisfied for the sensitive groups men and women. We can make the (weighted) positive prediction rates the same by adjusting e_{23}'s weight from 2 to 1. As a result, the (weighted) positive prediction rates for men and women have the same value of $\frac{1.0}{1.0+1.0} = 0.5$.

In reference [33], we compare MLClean with other baselines that use a strict subset of the operations data sanitization, data cleaning, and unfairness mitigation or use all three operations, but in a different order than MLClean. On real datasets, MLClean has the best model accuracy and fairness, demonstrating that all three operations are necessary for the best results. In addition, MLClean is faster than baselines that use the three operations in different orders, which means that utilizing the dependencies among the operations is important.

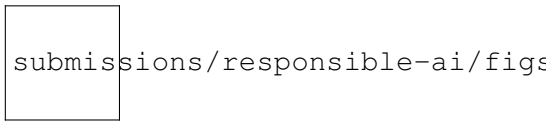

Figure 7: MLClean running on our examples.

5 Usable Responsible AI

The final pillar of responsible AI is making it usable and actionable to all machine learning users. While usability is not always the main focus in machine learning, it is especially relevant for responsible AI because the various objectives are already conceptually challenging to understand, so the deployment must be made as easy as possible. We thus propose two systems: FairBatch [29] is an easy-to-use model training technique for fairness, and Slice Finder [13, 14] is an easy-to-use model evaluation technique for improving fairness.

5.1 Batch Selection for Fair Models

While many unfairness mitigation techniques [9] have been proposed, most of them require significant amounts of effort to deploy. Pre-processing techniques have the advantage of being applicable to any model, but require

changes in the training data in order to remove bias. In-processing techniques tend to perform well, but usually propose a new model training algorithm that completely replaces an existing algorithm. An interesting question is whether we can take the best of both worlds of pre-processing and in-processing without their overheads.

We show that such a solution exists and propose FairBatch, which simply improves the batch selection of stochastic gradient descent training for better fairness. We formulate a bilevel optimization problem where we keep the standard training algorithm as the inner optimizer while incorporating the outer optimizer to equip the inner problem with the additional functionality: adaptively selecting minibatch sizes for the purpose of improving fairness. While the model is training, FairBatch adaptively adjusts the portions of the sensitive groups within each batch that is selected for each training epoch based on the fairness of the current intermediate model. For example, let us use the COMPAS example where we are predicting recidivism rates of criminals. Also let us use equalized odds (defined in Section 3) as the fairness measure where we want the positive prediction rates of sensitive groups to be the same conditioned on the true label. Since the label is fixed, this fairness can be interpreted as the model having the same accuracy for sensitive groups conditioned on the label. Now suppose that an intermediate model shows higher accuracy for a certain sensitive group. FairBatch then increases the batch-size ratio of the other underperforming sensitive group in the next batch. Intuitively, a larger batch size ratio results in better accuracy, so eventually equalized odds will improve. Figure 8 (a) illustrates how FairBatch improves equalized odds during a single model training. In reference [29], we show that this strategy is theoretically justified and generalize the algorithm for other fairness measures including demographic parity.

A key feature of FairBatch is its usability where one only needs to replace the batch selection of a machine learning system. Figure 8 (b) shows a PyTorch code example where one can deploy FairBatch by replacing a single line of code, and no further changes are needed in the pre-processing or in-processing steps of model training.

In reference [29], we also conduct experiments on synthetic and real datasets and show that FairBatch surprisingly has performances comparable to or even better than state-of-the-art pre-processing and in-processing unfairness mitigation techniques in terms of accuracy, fairness, and runtime. In addition, FairBatch is flexible and can be used to improve the fairness of pre-trained models like ResNet18 and GoogLeNet. Finally, there are batch selection techniques proposed for faster model training convergence, and FairBatch can be naturally combined with them to improve fairness as well.

5.2 Automatic Data Slicing for Fair Models

After model training, models are evaluated before being served. For example, TensorFlow Model Analysis [4] is a model evaluation component of TFX that accepts a user-specified slicing feature (e.g., country) and shows the model accuracies per slice (e.g., accuracy per country). Here we are using equalized error rates (defined in Section 4.1) as our notion of fairness. However, there is potentially an exponential number of slices to explore, and it is not easy for users who do not have enough domain expertise to quickly sift through them.

We thus propose Slice Finder [13, 14], which automatically finds "problematic" slices (subsets of the data) where the model underperforms. Given these slices, users can take action by acquiring more data as in Slice Tuner or debug the problematic data to find the root cause that led to the poor performance. We define a problematic slice to have the following characteristics. First, the slice must be interpretable where it can be defined with feature-value pairs, e.g., "Gender=Male and Age=20-30." While one can also define a slice to be a cluster of examples, clusters are often difficult to understand in practice. In addition, the slice must have a relatively lower accuracy than its complement, i.e., the rest of the examples other than the slice, where the difference (effect size) is large and statistically significant. Finally, the slice must be large enough to have a meaningful impact on the overall model accuracy.

Since the search space for all possible slices is vast, we propose two approaches for searching. The first is a decision tree approach where we construct a decision tree of feature-value pairs to find slices. The traversal is fast, but the slices are non-overlapping, which means that we may miss some problematic slices. The second

is a lattice search approach where we find slices by traversing a lattice of feature-value pairs in a breadth-first manner. Although we now find overlapping slices, this searching is slower than the decision tree approach. Once we find potential problematic slices, we perform effect-size and significance testings.

In references [13, 14], we show that Slice Finder performs better than a clustering baseline on real datasets. Also while lattice searching is slower than decision tree searching, it finds more problematic slices.

6 Open Challenges

We are far from achieving responsible AI for end-to-end machine learning and suggest promising directions. First, there needs to be deeper and broader support for the responsible AI objectives in each step of end-to-end machine learning. In addition, we believe the usability aspect of responsible AI has been largely understudied, and that there needs to be more emphasis on this important direction. Below are some concrete suggestions.

- *Data Collection*: We believe data acquisition must also support robustness. Dataset searching is becoming increasingly easy, and one challenge is distinguishing any poisoned data from the rest of the data. We also believe it is important to address fairness and robustness in data labeling.
- *Data Cleaning and Validation*: MLClean is preliminary, and an interesting direction is to develop more general and automatic cleaning and validation techniques that support various combinations of data cleaning algorithms, fairness measures, and poisoning attacks.
- *Model Training*: FR-Train is a first of its kind and can be extended in many ways. First, there needs to be more investigation on how to defend against more sophisticated poisoning attacks other than labeling flipping. Second, algorithm stability is a well-known issue in adversarial training and can be improved. Third, one may want to train models without a clean validation set.
- *Model Evaluation*: There needs to be more robustness research for model evaluation where we can easily tell whether a model is accurate enough despite data poisoning in the training data.
- *Model Management and Serving*: There needs to be more model managing and serving techniques that support fairness and robustness. While there are task-specific solutions like fairness in ranking [32], an interesting direction is to generalize and support any task with minimal configuration.
- There needs to be holistic solutions for the rest of the responsible AI objectives including explainability, transparency, and accountability. For example, recent data provenance and metadata [1, 10] solutions can be used to explain why each step in machine learning produced a certain result.

7 Conclusion

We proposed three research directions – depth, breadth, and usability – towards fully supporting responsible AI in end-to-end machine learning. While most research focuses on supporting one of many responsible AI features, we believe multiple objectives should be supported together, preferably in all steps from data collection to model serving. So far, we have scratched the surface of this vision where we proposed the following systems: FR-Train (holistic fair and robust training), Slice Tuner (selective data acquisition for fair models), MLClean (data cleaning for fair and robust models), FairBatch (easy-to-use batch selection for fair models), and Slice Finder (easy-to-use problematic slice finding for fair models). We also suggested various open challenges.

Acknowledgement

This work was supported by a Google AI Focused Research Award and by the Engineering Research Center Program through the National Research Foundation of Korea (NRF) funded by the Korean Government MSIT (NRF-2018R1A5A1059921).

References

[1] Machine learning metadata. https://www.tensorflow.org/tfx/guide/mlmd. Accessed Jan 15th, 2021.

[2] Responsible ai practices. https://ai.google/responsibilities/responsible-ai-practices. Accessed Jan 15th, 2021.

[3] Responsible ai principles from microsoft. https://www.microsoft.com/en-us/ai/responsible-ai. Accessed Jan 15th, 2021.

[4] Tensorflow model analysis. https://www.tensorflow.org/tfx/guide/tfma. Accessed Jan 15th, 2021.

[5] Trusting ai. https://www.research.ibm.com/artificial-intelligence/trusted-ai/. Accessed Jan 15th, 2021.

[6] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias: There's software used across the country to predict future criminals. And its biased against blacks., 2016.

[7] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning*. fairmlbook.org, 2019. http://www.fairmlbook.org.

[8] Denis Baylor, Eric Breck, Heng-Tze Cheng, Noah Fiedel, Chuan Yu Foo, Zakaria Haque, Salem Haykal, Mustafa Ispir, Vihan Jain, Levent Koc, Chiu Yuen Koo, Lukasz Lew, Clemens Mewald, Akshay Naresh Modi, Neoklis Polyzotis, Sukriti Ramesh, Sudip Roy, Steven Euijong Whang, Martin Wicke, Jarek Wilkiewicz, Xin Zhang, and Martin Zinkevich. TFX: A tensorflow-based production-scale machine learning platform. In *KDD*, pages 1387–1395, 2017.

[9] Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John T. Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. AI fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM J. Res. Dev.*, 63(4/5):4:1–4:15, 2019.

[10] Emily M. Bender and Batya Friedman. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *TACL*, 6:587–604, 2018.

[11] Omar Benjelloun, Shiyu Chen, and Natasha F. Noy. Google dataset search by the numbers. In *ISWC*, pages 667–682, 2020.

[12] Eric Breck, Martin Zinkevich, Neoklis Polyzotis, Steven Euijong Whang, and Sudip Roy. Data validation for machine learning. In *MLSys*, 2019.

[13] Yeounoh Chung, Tim Kraska, Neoklis Polyzotis, Ki Hyun Tae, and Steven Euijong Whang. Slice finder: Automated data slicing for model validation. In *ICDE*, pages 1550–1553, 2019.

[14] Yeounoh Chung, Tim Kraska, Neoklis Polyzotis, Ki Hyun Tae, and Steven Euijong Whang. Automated data slicing for model validation: A big data - AI integration approach. *IEEE TKDE*, 32(12):2284–2296, 2020.

[15] Gabriela F. Cretu, Angelos Stavrou, Michael E. Locasto, Salvatore J. Stolfo, and Angelos D. Keromytis. Casting out demons: Sanitizing training data for anomaly sensors. In *IEEE S&P*, pages 81–95, 2008.

[16] Tobias Domhan, Jost Springenberg, and Frank Hutter. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *IJCAI*, pages 3460–3468, 2015.

[17] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *KDD*, pages 259–268, 2015.

[18] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *NeurIPS*, pages 3315–3323, 2016.

[19] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory F. Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *CoRR*, abs/1712.00409, 2017.

[20] Ihab F. Ilyas and Xu Chu. *Data Cleaning*. ACM, 2019.

[21] Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowl. Inf. Syst.*, 33(1):1–33, 2011.

[22] Pang Wei Koh, Jacob Steinhardt, and Percy Liang. Stronger data poisoning attacks break data sanitization defenses. *CoRR*, abs/1811.00741, 2018.

[23] Arun Kumar, Matthias Boehm, and Jun Yang. Data management in machine learning: Challenges, techniques, and systems. In *SIGMOD*, pages 1717–1722, 2017.

[24] Andrea Paudice, Luis Muñoz-González, and Emil C. Lupu. Label sanitization against label flipping poisoning attacks. In *ECML PKDD*, pages 5–15, 2018.

[25] Neoklis Polyzotis, Sudip Roy, Steven Euijong Whang, and Martin Zinkevich. Data management challenges in production machine learning. In *SIGMOD*, pages 1723–1726, 2017.

[26] Neoklis Polyzotis, Sudip Roy, Steven Euijong Whang, and Martin Zinkevich. Data lifecycle challenges in production machine learning: A survey. *SIGMOD Rec.*, 47(2):17–28, 2018.

[27] Yuji Roh, Geon Heo, and Steven Euijong Whang. A survey on data collection for machine learning: a big data - AI integration perspective. *IEEE TKDE*, 2019.

[28] Yuji Roh, Kangwook Lee, Steven Euijong Whang, and Changho Suh. Fr-train: A mutual information-based approach to fair and robust training. In *ICML*, volume 119, pages 8147–8157, 2020.

[29] Yuji Roh, Kangwook Lee, Steven Euijong Whang, and Changho Suh. Fairbatch: Batch selection for model fairness. In *ICLR*, 2021.

[30] Sebastian Schelter, Yuxuan He, Jatin Khilnani, and Julia Stoyanovich. Fairprep: Promoting data to a first-class citizen in studies on fairness-enhancing interventions. In *EDBT*, pages 395–398, 2020.

[31] Hwanjun Song, Minseok Kim, Dongmin Park, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *CoRR*, abs/2007.08199, 2020.

[32] Julia Stoyanovich, Bill Howe, and H. V. Jagadish. Responsible data management. *PVLDB*, 13(12):3474–3488, 2020.

[33] Ki Hyun Tae, Yuji Roh, Young Hun Oh, Hyunsu Kim, and Steven Euijong Whang. Data cleaning for accurate, fair, and robust models: A big data - AI integration approach. In *DEEM@SIGMOD*, 2019.

[34] Ki Hyun Tae and Steven Euijong Whang. Slice tuner: A selective data acquisition framework for accurate and fair machine learning models. In *SIGMOD*, 2021.

[35] Suresh Venkatasubramanian. Algorithmic fairness: Measures, methods and representations. In *PODS*, page 481, 2019.

[36] Steven Euijong Whang and Jae-Gil Lee. Data collection and quality challenges for deep learning. *Proc. VLDB Endow.*, 13(12):3429–3432, 2020.

[37] Matei Zaharia, Andrew Chen, Aaron Davidson, Ali Ghodsi, Sue Ann Hong, Andy Konwinski, Siddharth Murching, Tomas Nykodym, Paul Ogilvie, Mani Parkhe, Fen Xie, and Corey Zumar. Accelerating the machine learning lifecycle with mlflow. *IEEE Data Eng. Bull.*, 41(4):39–45, 2018.

[38] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *AIES*, pages 335–340, 2018.

[39] Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *CVPR*, pages 4352–4360, 2017.

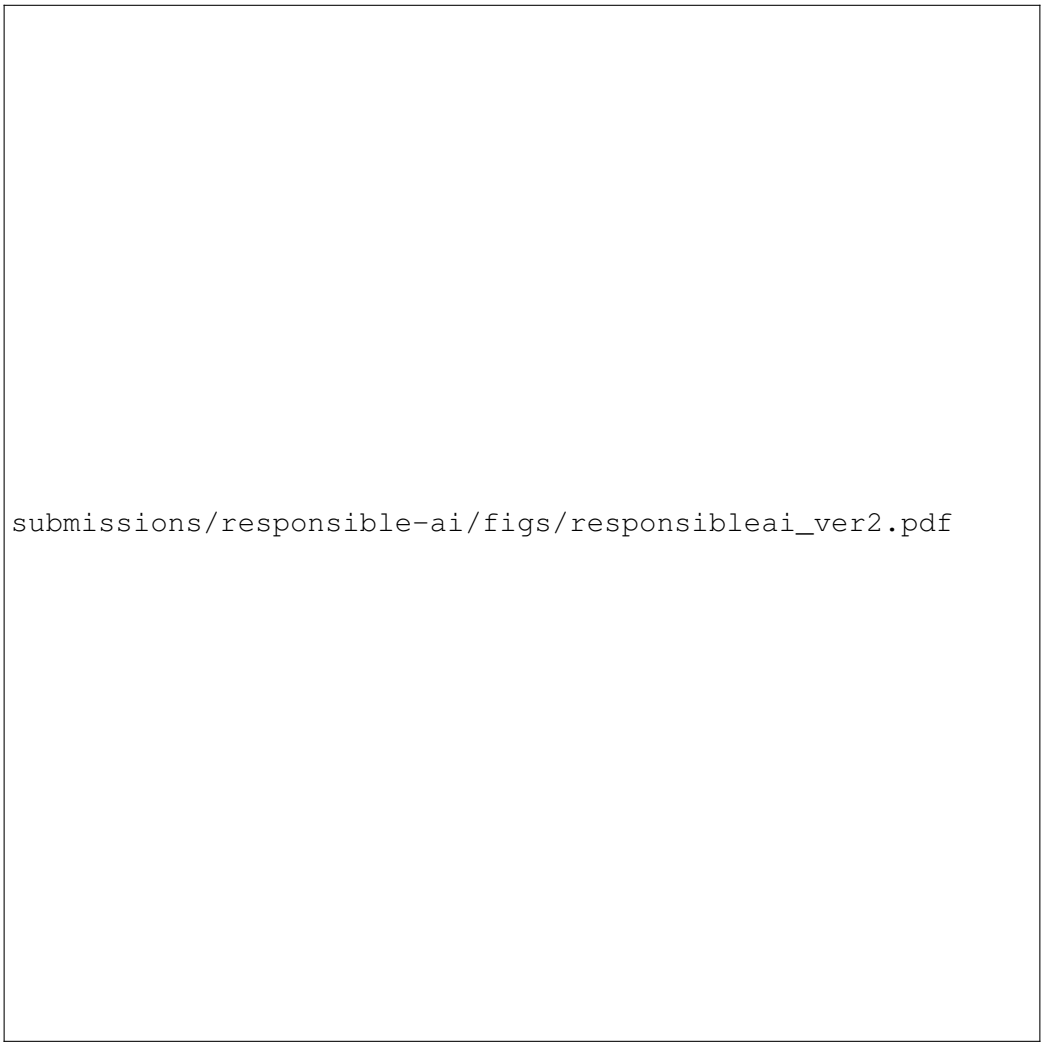

Figure 1: The three research directions – depth, breadth, and usability – for fully supporting the responsible AI objectives (fairness, robustness, and others) in addition to accuracy in end-to-end machine learning. The highlighted parts show our contributions: Slice Tuner [34] addresses fairness in data collection; MLClean [33] addresses fairness and robustness in data cleaning; FR-Train [28] addresses fairness and robustness in model training; FairBatch [29] addresses usability for fairness in model training; and Slice Finder [13, 14] addresses usability for fairness in model evaluation.

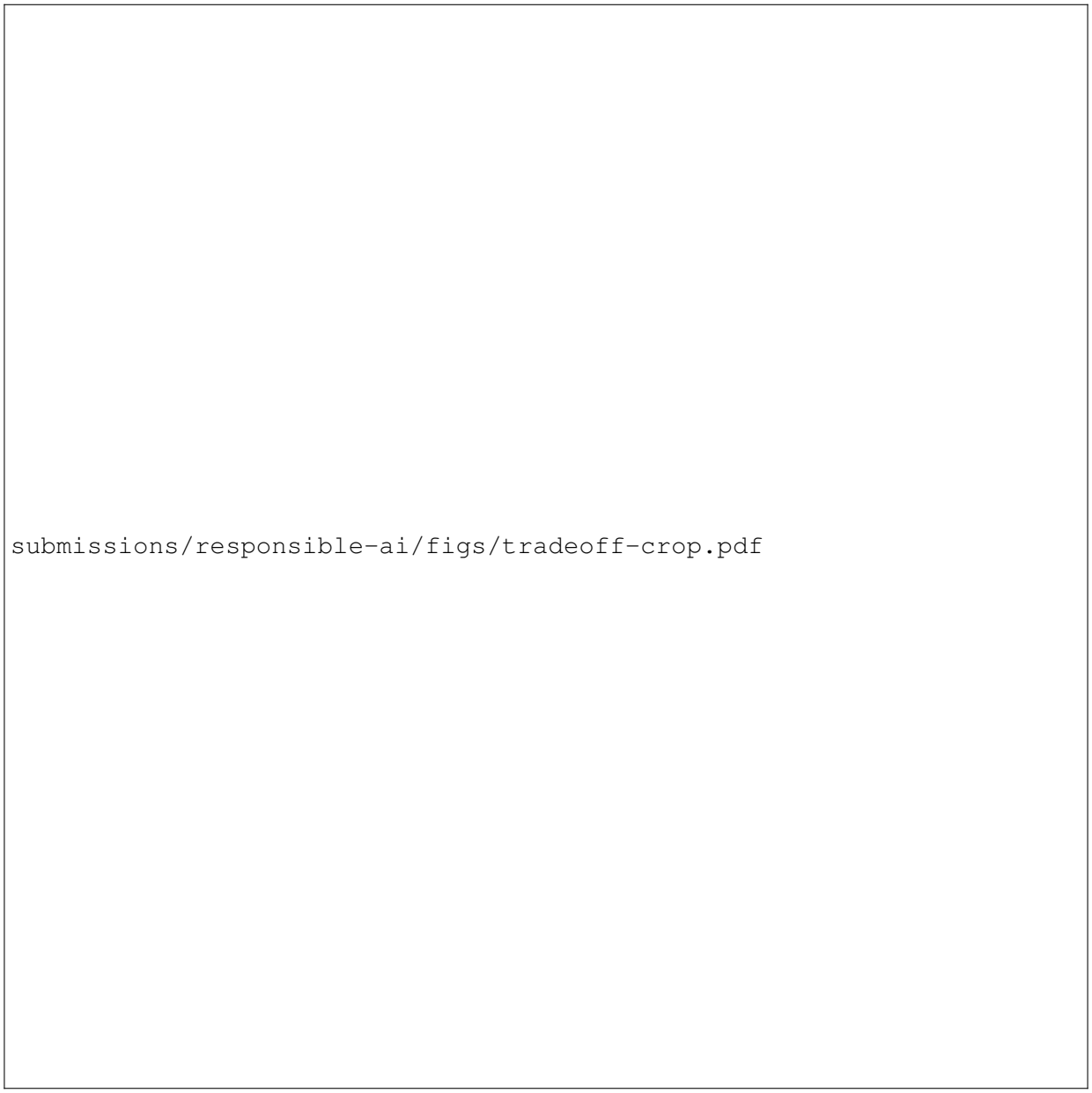

Figure 2: (Left) Accurate (black solid line) and fair (blue dotted line) classifiers on clean data followed by data poisoning. (Right) A fair classifier trained on poisoned data (red dotted line) is evaluated on clean data, showing a worse accuracy-fairness tradeoff than the fair classifier trained on clean data.

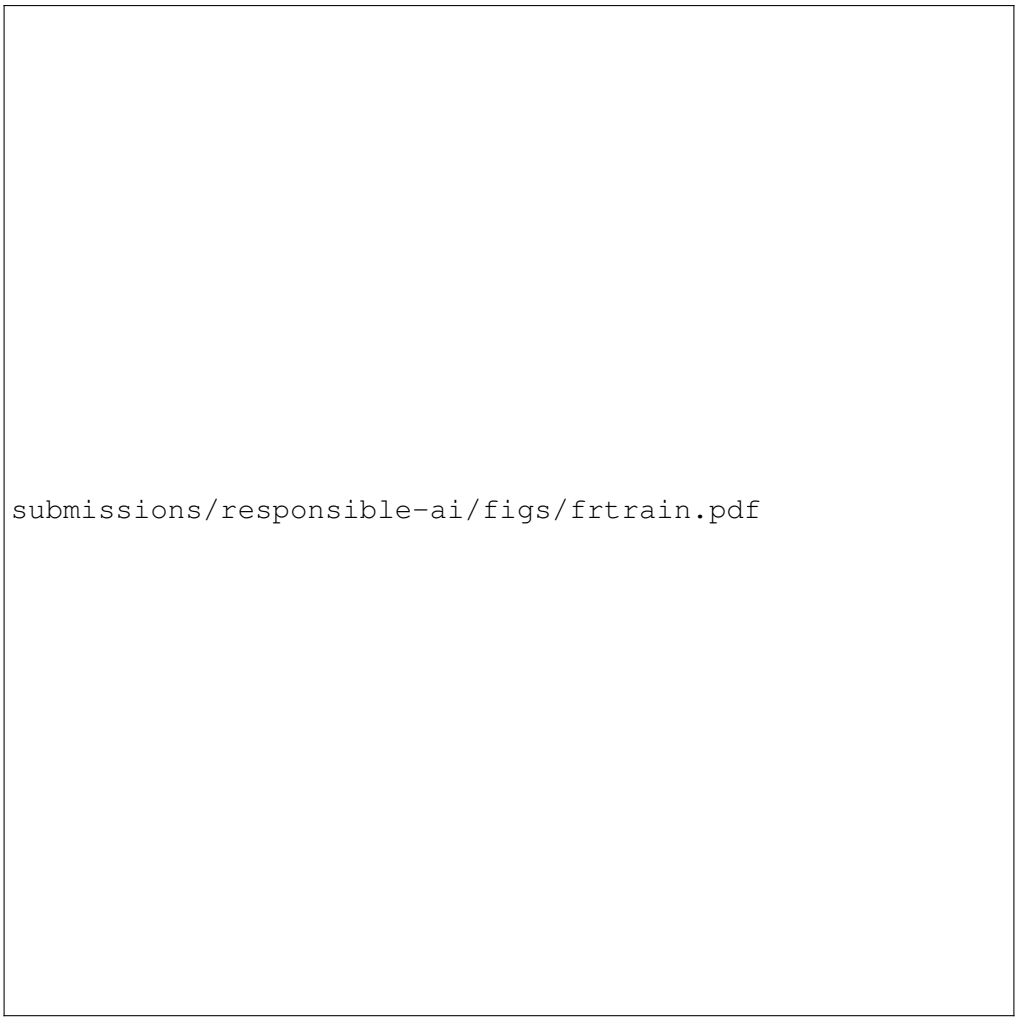

Figure 3: The FR-Train architecture and how it can be used for recidivism prediction.

Figure 4: (a) Learning curves on two slices of the UTKFace dataset [39]. (b) Slice Tuner architecture.

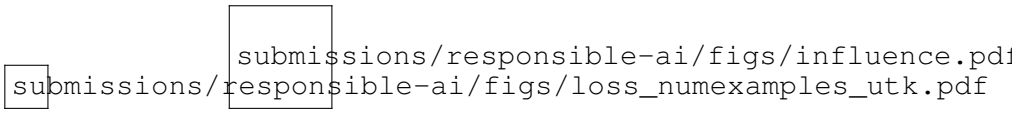

Figure 5: (a) Data acquisition on the slice White-Male influencing the losses on the other slices for the UTKFace dataset. (b) To give an intuition, say there are three slices where shape indicates slice, and color indicates label. (Top) If we only increase the triangles, the decision boundary may shift to the left due to the new bias, changing the losses of the other slices. (Bottom) If we evenly increase the data for all slices, the bias does not change, and there is little influence among the slices.

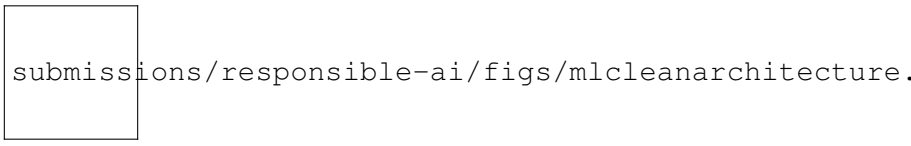

Figure 6: The MLClean architecture where data sanitization and data cleaning are performed together followed by unfairness mitigation. Optionally, a previous model can be used by any of the three operations.

Figure 8: (a) The black path shows how the model fairness improves as FairBatch adjusts two parameters λ_1 (sampling rate for examples where Z=0 given Y=0) and λ_2 (sampling rate for examples where Z=0 given Y=1) for each epoch on the COMPAS dataset. "ED disparity" is the accuracy difference conditioned on the true label between sensitive groups where lower disparity means better equalized odds. (b) Sample PyTorch code where the batch selection sampler is replaced with FairBatch with a single-line change highlighted in blue.