Leave-one-out Unfairness

Emily Black
emilybla@andrew.cmu.edu
Carnegie Mellon University

Matt Fredrikson
mfredrik@cs.cmu.edu
Carnegie Mellon University

ABSTRACT

We introduce *leave-one-out unfairness*, which characterizes how likely a model's prediction for an individual will change due to the inclusion or removal of a *single* other person in the model's training data. Leave-one-out unfairness appeals to the idea that fair decisions are not arbitrary: they should not be based on the chance event of any one person's inclusion in the training data. Leave-one-out unfairness is closely related to algorithmic stability, but it focuses on the consistency of an individual point's prediction outcome over unit changes to the training data, rather than the error of the model in aggregate. Beyond formalizing leave-one-out unfairness, we characterize the extent to which deep models behave leave-one-out unfairly on real data, including in cases where the generalization error is small. Further, we demonstrate that adversarial training and randomized smoothing techniques have opposite effects on leave-one-out fairness, which sheds light on the relationships between robustness, memorization, individual fairness, and leave-one-out fairness in deep models. Finally, we discuss salient practical applications that may be negatively affected by leave-one-out unfairness.

ACM Reference Format:
Emily Black and Matt Fredrikson. 2021. Leave-one-out Unfairness. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT '21), March 3–10, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3442188.3445894

1 INTRODUCTION

Deep networks are becoming the go-to choice for challenging classification tasks due to their remarkable performance on many high-profile problems: they are used everywhere from recommendation systems [15] to medical research [8, 21], and increasingly in even more sensitive contexts, such as hiring [46], loan decisions [5, 51], and criminal justice [25]. Their continued rise in adoption has led to growing concerns about the tendency of these models to discriminate against certain individuals [4, 10, 13, 44], or otherwise produce outcomes that are seen as unfair.

There are several definitions that aim to formalize fair behavior in machine learning contexts: group-based notions, such as demographic parity [23] and equalized odds [26], stipulate that different demographic groups should be treated similarly in aggregate; on the other hand, individualized notions focus on how each person is treated, such as individual fairness [20], which requires "similar" outcomes for similar people, and counterfactual fairness [34], which argues that people should be treated the same as their hypothetical counterpart, who takes a different protected attribute. Fundamentally, *these fairness criteria depend on a comparison of how one group or individual is treated versus another.* However, there are also situations where the decision-making mechanism is unfair not because of how its behavior varies across defined groups or individuals, but rather because its decisions cannot be justified by consistent, intelligible criteria. In other words, decisions may be unfair because they are arbitrary.

In this paper, we study the extent to which instability can lead to such fairness issues. Intuitively, when a person's outcome hinges on the presence of another, single individual in the training data, the outcome that follows may be viewed as unfair. Take for example a person in reasonable financial health who applies for an auto loan. Suppose that whether their application is approved or not depends on whether another *unrelated* person had applied for a loan from the same bank, and was subsequently included in the training data. Such a decision may be viewed as unfair, as it depends on the willingness and availability of another person to provide their data for training—a chance occurrence, rather than a well-justified set of criteria. Even beyond its potential unfairness, this behvaior may be especially undesireable in applications which come with a "right to explanation" [33].

Measuring leave-one-out Unfairness. To formalize this intuition, we introduce *leave-one-out unfairness* (LUF): the chance that an individual's outcome will change due to the presence of any one instance in the training data (Section 3, Definition 2). To the best of our knowledge, this is the first attempt to formalize unfairness as stemming from the *arbitrary* nature of decision rules, and in particular the stability of the underlying learning algorithm. Certainly, there are other random choices made during model development that may lead to an arbitrary change in model outcome for an individual—changes in the random initialization or architecture, for example, which we explore in Section 6. However, we focus on instability with respect to training data in particular due to its connections to other areas of machine learning literature such as stability, privacy, and robustness.

We find that in many cases, the use of deep models can lead to this type of unfair outcome with surprising frequency, and can result in different outcomes for seemingly unrelated individuals. To gain an intuition for why this might be, Figure 1 depicts the decision boundaries of two low-dimensional binary classifiers whose training data differs only on the presence of the point highlighted in red. Notice that the boundary near the left-out point remains fairly consistent, but there are non-trivial differences in both the boundary locations and the confidence of the model's predictions in regions away from the point. While this low-dimensional example provides some intuition, we systematically characterize the extent to which deep models behave as such on real data (Section 4). We find that it occurs often enough to be a concern in some settings (i.e., up to 7% of data is affected); that it occurs even on points for which the model assigns high confidence; and is not consistently influenced by dataset size, test accuracy, or generalization error (Figure 4, Table 2).

FAccT '21, March 3–10, 2021, Virtual Event, Canada
ACM ISBN 978-1-4503-8309-7/21/03.
https://doi.org/10.1145/3442188.3445894

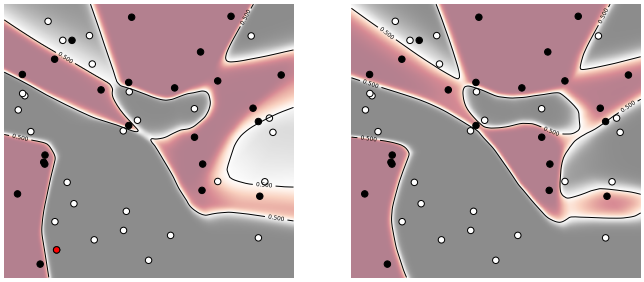

Figure 1: Classification boundaries of a deep model with three hidden layers, trained on two-dimensional data with uniform-random binary labels, before (left) and after (right) the point highlighted in red is removed from the training data. Lighter regions correspond to predictions with less confidence. While the model remains largely unchanged in the area around the left-out point, its boundary changes significantly in other, far-away areas. For example, the middle-right region assigns greater confidence to white points, even flipping its prediction on one such point.

Connections. Leave-one-out unfairness has useful connections to other fields such as stability, privacy, and robustness. We show that while LUF is strictly stronger than some prior notions of leave-one-out stability [49] (Section 3.3, Proposition 3.2), it is *weaker* than differential privacy [18] (Proposition 3.3). Thus, one can achieve bounded levels of leave-one-out unfairness by satisfying differential privacy, but it may also be possible to do so via relaxations that allow greater flexibility in the selection of learning rules [41].

Recent work has related robust classification to desirable properties beyond mitigating adversarial examples [54], such as the encoding of more human-interpretable features [22, 31, 43, 56], and individual fairness on weighted ℓ_p metrics [61]. These results may seem to suggest that robust models would also be less susceptible to leave-one-out unfairness. Evaluating two common techniques for producing robust models, adversarial training [40] and randomized smoothing [14], we find that these methods in fact have *vastly different* effects on leave-one-out unfairness. Whereas randomized smoothing tends to have no effect, adversarial training amplifies the problem, resulting in up to a factor of five more affected points (Section 5). These results suggest that although LUF and robustness are not inherently tied to each other, certain types of models may prove beneficial for both.

Summary. In a similar vein to the oft-cited "lack of interpretability" [38], leave-one-out unfairness complicates the responsible application of deep models to sensitive decisions. Particularly in settings where a "right to explanation" is pertinent [33], these complications may need to be weighed against the benefits that deep models provide over less complex alternatives. This paper presents the first steps towards a better understanding of this issue, and points to several intriguing directions for future study. To summarize, we present the following contributions:

(1) We introduce and formalize *leave-one-out unfairness*, which characterizes a possible source of unfair, arbitrary outcomes in ML applications.
(2) We relate leave-one-out unfairness to well-known prior notions of stability, shedding light on when models may suffer from leave-one-out unfairness, and techniques that might help to mitigate it.
(3) Finally, we present an extensive evaluation of how prevalent LUF is when deep neural networks are trained on a variety of datasets, and compare it to other sources of instability such as random initialization and choice of architecture.

Figure 2: From left to right: Individual removed from the dataset (z). When z is included in the training set, the two individuals to the right (x, y) are labeled as a match with confidence 0.84. When z is not in the dataset, x and $y$are predicted as *not* a match with confidence 0.07.

In Section 2, we provide two examples of machine learning applications where leave-one-out unfairness may lead to unjust model behavior, along with experimental results demonstrating that LUF indeed may occur in these contexts. Following this, in Section 3, we formally define leave-one-out unfairness and explore its relationships to LOO-stability and differential privacy. In Section 4 and Section 5, we present our experimental results of the extent of leave-one-out unfairness on real datasets for conventional and robustly trained machine learning models.

2 CONTEXTUALIZING LEAVE-ONE-OUT UNFAIRNESS

Leave-one-out unfairness may not pose a problem in all machine-learning applications. If the model's outcome is of little consequence to peoples' lives, or if the application context does not require consistency across data samples for adequate justification, then arbitrary predictions may be acceptable. Determining whether or not leave-one-out unfairness leads to fairness issues requires considering this context. In this section, we motivate examples of how leave-one-out unfairness constitutes a fairness issue in two contexts: facial recognition use by law enforcement, and loan application decision models used by financial institutions.

2.1 Facial Recognition

Facial Recognition Technology (FRT) has proliferated in recent years as a method of verifying identity at scale. Its use in law enforcement, and the potential harms that may follow, have gained particular attention due to the potentially dire consequences of misidentification: matches for facial recognition matches have been used as evidence for arrest [29, 57]. Moreover, the use of this technology in this context is becoming prevalent: according to a study from 2016 [25], at least one in four police agencies in the United States have made use of it.

Background. The use of FRT by law enforcement relies primarily on *face-matching* models, where two face images are provided as input to determine whether they depict the same individual. Note that this differs from *face classification* models, which aim to identify the person depicted in a face image from a pre-determined set of

individuals. A typical workflow proceeds as follows: given an image of a suspect, law enforcement queries a face-matching model against a large set of images in a database, which also contains identifying information. The face-matching model provides a binary label, with a confidence score, and the most confident matches are provided to the operator for further review [47].

Many police agencies use ready-made, third-party models. For example, one such third-party, Clearview AI, reportedly contracts with approximately 2,400 law enforcement agencies [39]. Such third-party models are often trained on images obtained from public sources like the Internet, in particular by taking advantage of Creative-Commons licenses widely used on social media websites. [42]. The database of images on which these models are run during inference are often obtained from public records such as drivers license databases. Notably, these databases may largely consist of individuals with no prior criminal record [25].

Impact of Instability. The results FRT are increasingly being used by law enforcement as evidence to justify arrest [29, 57]. According to U.S. law, an individual must be arrested for a justifiable reason, i.e. probable cause [1]: a police officer must have evidence leading them to believe that the person arrested likely did commit the crime in question. Thus, when FRT results are cited when justifying probable cause, the factors that lead a particular face-matching model to its predictions must be scrutinized. In particular, if it is likely that a matching outcome can change due to the inclusion of a particular image–unrelated to the suspect or the potential match—out of tens of thousands in the model's training set, then it may be argued the evidence used to justify the eventual arrest is based on a chance occurrence, rather than on convincing facts relevant to the case. In short, such an outcome would be unfair due to the arbitrary nature of the supporting evidence. We aim to formalize this behavior, and investigate its prevalence on models trained on real datasets, including face-matching models.

Experimental Confirmation. We trained a face-matching model on Labeled Faces in the Wild (LFW) [30], consisting of 13,000 unconstrained pictures of 1680 different individuals. To measure the effect of individual images on prediction outcomes, we trained models both with and without a randomly sampled individual, controlling for all sources of non-determinism (e.g., parameter initialization and GPU operations). We repeated this experiment for 25 different randomly sampled individuals, and measured the effects on prediction behavior. Further details of our methodology are given in Section 4.

We found that the predictions given by the face-matching model change across datasets with single-image differences, with surprising frequently. One such example of this behavior is shown in Figure 2. When person z is included in the dataset, persons x and y are labeled as a match; but when person z is removed, they are not. Persons x and y are clearly different from one another, and aside from gender, share few salient characteristics. More surprisingly, both predictions are made with high confidence—0.84 and 0.07–far from a baseline random guess. Such behavior was not limited to these images, but rather we observed that 12% of the model's predictions changed across datasets differing in one image, while the change in accuracy remained less than 2%. Moreover, this behavior was consistent across changes in random initialization and choice of architectures, including a residual network resembling ResNet50.

	age	*education*	*occupation*	*sex*	*capital gain*	*model conf.*
Affected point (x)	51	Bachelors	Self-employed	F	0	0.87
LOO point (z)	39	11th Grade	Service Industry	M	0	-

Table 1: Selected feature values for a point treated leave-one-out unfairly in a deep model on the Adult dataset, and the point z whose removal resulted in the change in prediction. Confidence refers to the raw output of the model's prediction in the model with z.

2.2 Consumer Finance

Machine learning is also finding uses in consumer finance [7, 9, 50, 51]. Not surprisingly, the predictions made by these models, too, can greatly impact peoples' lives, potentially playing a decisive role in their ability to obtain buy a car, a house, or start a business.

Impact of Instability. Models used in this context may be expected to have consistent, justifiable reasons for the predictions that they make. A salient example is credit models used to inform lending decisions, where in Europe the General Data Protection Regulation (GDPR) requires that creditors using automated decision systems release "meaningful information about the logic involved" to applicants [3]. Similar regulations are relevant in the US through the Federal Deposit Insurance Corporation (FDIC) consumer protection law [2], which provides a "right to explanation" in lending decisions.

Some interpretations argue that the right to explanation provided by the GDPR requires that it should be possible to trace a decision back to pertinent details of an individual's loan application, and further that "the information about the logic must be meaningful to [the applicant], notably, a human and presumably without particular technical expertise" [48]. This suggests that if the explanation is not legible to the applicant based on prevailing norms, e.g. if it seems to be made based on incomprehensible or arbitrary facts such as the incidental makeup of the model's training data, then such a decision infringes upon their "rights and freedoms". After receiving an explanation, the GDPR provides the applicant the right to contest such a decision, and request human review.

Experimental Confirmation. As with the face-matching model in the previous subsection, we conducted experiments on models trained to predict a proxy for creditworthiness using datasets differing in a single instance. We used the UCI Adult dataset [17], consisting of a subset of US census data, and trained one-hidden-layer neural networks with 200 internal units to predict income from demographic, education, and employment information (details in Section 4). Our results suggest that the predictions of these models are often sensitive to the presence of single instances, indicating the potential for *leave-one-out unfairness*.

Looking more closely at the results, one of these models was trained with the point z shown in Table 1 included in the training set: a 39-year-old man with an 11th-grade education who works in the service industry. This model predicts that a 51-year-old, college-educated, self-employed woman makes more than $50k (0.87 confidence), whereas a model trained on the same data *without* z made the opposite prediction. Mirroring our findings with the FRT models, there is no apparent connection between the features that represent these individuals (see Table 1), and the models predict the woman's outcome with high confidence. The removal of this one individual does not just affect this 51-year-old woman, but rather we find that approximately 2% of the entire data set, 603 predictions, are changed.

3 LEAVE-ONE-OUT UNFAIRNESS

In this section, we introduce the definition of leave-one-out unfairness, and discuss its connections to prior notions of stability: leave-one-out stability [49], differential privacy [18], and individual fairness [20]. We prove that leave-one-out unfairness is a stronger notion than leave-one-out stability, and weaker than differential privacy. Our formalization of LUF allows us to measure its prevalence objectively on real data, and our investigation of its connections to other forms of stability suggest mitigation techniques as well potential middle ground for achieving gains in privacy.

3.1 Notation and Preliminaries

We assume a typical supervised learning setting. Let $z = (x,y) \in \mathbf{X} \times \mathbf{Y}$ be a data point, where x represents a set of features and y a response. Points z are drawn from a distribution $\mathcal{D}$, as are datasets S from the iid product of $\mathcal{D}$, i.e. $S \sim \mathcal{D}^n$. We assume that learning rules h are randomized mappings from datasets S to models h_S, which are functions mapping features to responses; in other words, $h_S : \mathbf{X} \to \mathbf{Y}$ is the model obtained by learning with h on data S. We use $U(m)$ to refer to the uniform distribution over the integers $\{1...m\}$. Given S sampled from $\mathcal{D}^n$ and index $i \sim U(m)$, we denote the sample S with the ith element removed as $S^{(\setminus i)}$.

3.2 Leave-one-out Unfairness

Leave-one-out unfairness is based on the notion that a model's treatment of an individual should not depend too heavily on the inclusion of any other single training point. This is related to the concept of algorithmic stability, which measures the effect that a small change in input has on an algorithm's output. For example, a machine learning algorithm is *stable* if a small change to its input (training set) causes limited change in its output (a trained model). Usually, the change in output is measured in the form of model error. Definition 1 formalizes this as *leave-one-out (LOO) stability*, but we note that there are several variants that quantify over pointwise *replacement* instead of leave-out, and use different types of aggregation in their bound [49].

Definition 1 (Leave-one-out (LOO) Stability [49]). *Let $\epsilon_{stable} : \mathbb{N} \to \mathbb{R}$ be a monotonically-decreasing function. Given a training set $S = (z_1,...,z_m) \sim \mathcal{D}^n$, and a training set $S^{(\setminus i)} = (z_1,...,z_{i-1},z_{i+1},...,z_m)$ with $i \sim U(m)$, a learning rule h is* leave-one-out-stable *(or* LOO-stable*) on loss function ℓ with rate $\epsilon_{stable}(m)$ if*

$$\frac{1}{m}\sum_{i=1}^{m} \mathop{\mathbb{E}}_{S\sim\mathcal{D}^n}\left[\left|\ell(h_S,z_i) - \ell(h_{S^{(\setminus i)}},z_i)\right|\right] \leq \epsilon_{stable}(m)$$

LOO-stability records the average effect of removing an individual from the training set on the absolute loss on that individual's prediction. Quantifying the effect model of instability on the fairness of predicted outcomes, however, calls for a definition focusing on different aspects of model behavior. LOO-stability is a predicate on a learning rule that can be satisfied in order to achieve an acceptable level of model stability, in expectation over all draws of a training set S. However, in this paper, we are interested in quantifying the extent of arbitrariness in a particular individual's prediction—to capture this, we need a *metric* of unfairness, rather than a fairness guarantee. Pursuant of capturing an particular individual's real-life experience with a particular model, we are interested in a quantifying arbitrary behavior in relation to a particular model context–i.e., on a fixed training set S.

To focus the effect of instability on the experience of the population on which it is deployed, rather than a measure of model performance, we need a metric which accounts for the instability that arises for *any* person from the inclusion of a given point in the training set—rather than the impact that the changed point has on the error its *own* prediction. Even with this focus on the experience of the individuals, an aggregate calculation such as in LOO-stability may hide the experiences of an unlucky few who may encounter particularly high arbitrariness in their outcome. To ensure that model behavior on every individual is considered, a worst-case metric is more suitable. Further, appealing to the intuition that a model acts unfairly if it is arbitrary, the *consistency* of its prediction, rather than its loss, is the target; consistent predictions, even when incorrect, suggest that the model's decision is not arbitrary. Definition 2, below, reflects these considerations.

Definition 2 (Leave-one-out Unfairness (LUF)). *Let D be the distribution from which the training set S is drawn, and let x be in the support of D. We define the leave-one-out unfairness (LUF) experienced by x under learning rule h and training set $S \sim D$ to be:*

$$\mathrm{LUF}(h,S,x) = \max_{i,k} \left|\Pr[h_S(x) = k] - \Pr[h_{S^{(\setminus i)}}(x) = k]\right|$$

The randomness in this expression is over the choices made by h. Note that in cases of a deterministic learning rule, $\Pr[h_S(x) = k]$ is 0 or 1.

In other words, given a learning rule h and a training set S, the LUF experienced by a person x is the worst-case probability that x receives a different prediction in a model trained with h on S, and one trained with h on S with a single point removed. Intuitively, this is one way of quantifying the arbitrariness of the model's decision at x. If LUF is high, then the model's decision is brittle under small, potentially irrelevant changes, i.e., a one-point change in the model's training set—casting doubt on the reason behind the model's decision.

In certain situations, such as when evaluating various models during development, it may be useful to understand the extent of leave-one-out unfairness across the entire population under a given learning rule: i.e. understanding how likely it is *any* individual in the distribution will experience an arbitrary decision. This motivates the concept of *expected* leave-one-out unfairness, defined below. As most of our experiments aim to measure the frequency and severity of arbitrary behavior across real datasets, we will focus most heavily on this definition throughout the paper.

Definition 3 (Expected Leave-one-out Unfairness). *Let D be the distribution from which the training set S is drawn, and let x be drawn randomly from D. We define the expected leave-one-out unfairness (LUF) experienced by x under learning rule h and training set $S \sim D$ to be:*

$$\mathbb{E}_x[\mathrm{LUF}(h,S,x)] = \mathbb{E}_{x\sim D}[\max_{i,k} \left|\Pr[h_S(x) = k] - \Pr[h_{S^{(\setminus i)}}(x) = k]\right|]$$

Where the randomness in the expectation is taken over samples of x from D.

3.3 Connections to Existing Stability Notions

While our introduction of Definition 2 above is clearly motivated by LOO stability, in this section we explore the connections to this and

other forms of stability in greater depth. Specifically, we demonstrate that while learning rules that are already known to be leave-one-out-stable may still be susceptible to leave-one-out unfairness, strategies for ensuring stronger notions of stability, such differential privacy, can be used to mitigate LUF. We also explore the connection between LUF and other individual-based fairness notions, i.e. individual fairness.

LOO Stability. Leave-one-out stability is a coarser notion than leave-one-out unfairness, as it records the average change in a model's error on a given point when that same point is removed from the training set. Meanwhile, LUF focuses on how a certain point's model outcome can change as a result of *any* point in the training set being removed.

A LOO-stable model may still treat points leave-one-out unfairly: a model can exhibit similar error *on a given point* before and after that point is removed from the training set, but it may treat other points differently. We demonstrate this point on the simple learning rule and distribution in Figure 3. Additionally, the fact LOO-stability is averaged over the entire training set can obscure the fact that some individual points are strongly affected by a small change in the training set. Proposition 3.1 formalizes this, showing that LOO-stability is strictly weaker than LUF.

Proposition 3.1. *Let h be a learning rule, ℓ be 0-1 loss, and $\epsilon(m)$ be a montonically-decreasing function such that h is leave-one-out stable with rate $\epsilon(m)$ for all $S \sim \mathcal{D}^m$. Then there exists a training set S such that $\mathbb{E}_x[\mathrm{LUF}(h,s,x)] > \epsilon_{stable}(m)$ and x.*

Proof. Consider a binary classification problem a discrete distribution D with three points, as pictured in Figure 3: $x_1, x_2 \in D$ are of class 0, and $x_3 \in D$ is of class 1, shown in red and blue. We define a learning rule, h, according to the different classifiers learned with each possible training set $S \sim D$, shown in Figure 3. Notice that this learning rule is LOO-stable with $\epsilon_{\text{stable}}(3)$=0, as when each point is removed, the classification error *on that point* remains the same: this is shown by construction in Figure 3 when $S = x_1, x_2, x_3$, and in all other cases, the learning rule is constant, as shown in the figure. Thus, $\frac{1}{3}\sum_{i=1}^{3} \mathbb{E}_{S\sim\mathcal{D}}[|\ell(h_S, z_i) - \ell(h_{S^{(\setminus i)}}, z_i)|] = 0 \leq 0$. However, notice that e.g., if $S = x_1, x_2, x_3$, and x_3 is removed, x_2 experiences a change in classification outcome. Thus, $\mathrm{LUF}(h,S,x_2) = 1$. See that, in fact, that every point is susceptible to a change in prediction as the result of different point being removed from the dataset—thus, $\mathbb{E}_x[\mathrm{LUF}(h,S,x)] = 1$. □

Proposition 3.2 shows that models with bounded LUF are also LOO-stable; the proof is given in the supplementary material.

Proposition 3.2. *Let h be a learning rule, ℓ be 0-1 loss, and $\epsilon(m)$ be a montonically-decreasing function such that $\mathrm{LUF}(h,S,x) \leq \epsilon(m)$ for all $S \sim \mathcal{D}^m$ and x. Then h is leave-one-out stable with rate $\epsilon(m)$.*

Differential Privacy. Privacy and fairness are related in various ways, as others have illustrated before [16, 20]. Like differential privacy, leave-one-out unfairness is a stability property of learning rules, but differential privacy is stronger. In particular, differential privacy (Definition 4) quantifies universally over all pairs of related training data, and limits the probability of any change in outcome. On the other hand, Definitions 2 and 3 fix a training set, and require stability of the model's response on points from the target distribution.

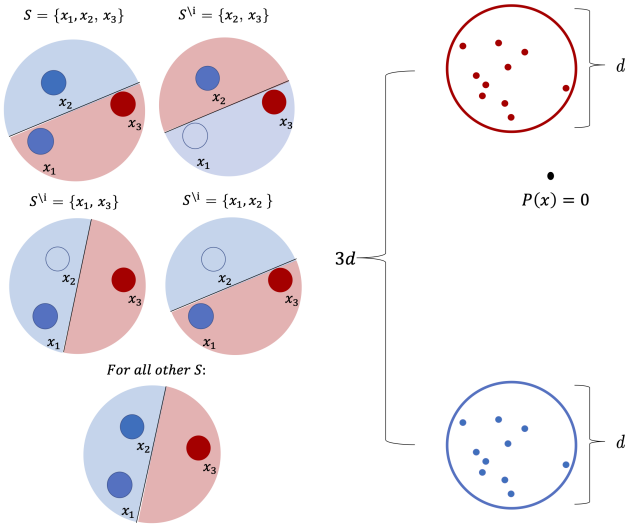

Figure 3: Left: A learning rule h that satisfies LOO-stability, but not expected LUF, over the distribution D of the three points pictured. In each box, we see the decision boundary learned with a specified training set $S \sim D$, thus fully defining h. The proof is explained in Proposition 3.1. Right: Visual intuition for how a model can have $LUF = 0, \forall x \in D$ but not satisfy differential privacy. Consider a 1-KNN model on a binary classification problem over the distribution pictured above: two perfectly separated uniform distributions over circles. The diameter of each circle is d, and the distance between the centers of the two circles is $3d$. Consider any training set S drawn from this distribution that has at least two data points from each class. See that $\mathrm{LUF}(h,S,x) = 0$ for all $x \in D$: removing any point from S cannot change the classification of any point *in the distribution*, i.e., within the circles pictured above. However, 1-KNN is not differentially private, as it is a deterministic, non-constant, learning rule. Specifically, see that adding or removing a point in S *can* shift the boundary sufficiently far to change the model's behavior on points *not* in D, (such as point x_2 pictured), which is a violation of differential privacy.

Definition 4 ((ϵ,δ) -Differential privacy). *An algorithm $A : \mathbf{X} \to \mathbf{Y}$ satisfies (ϵ,δ)-differential privacy, for $0 < \epsilon$ and $\delta \in [0,1]$, if for all $S \in \mathbf{X}^n$, $S' \in \mathbf{X}^{n-1}$ that differ in a single row and all $Y \subseteq \mathbf{Y}$, $\Pr[A(S) \in Y] \leq e^{\epsilon}\Pr[A(S') \in Y] + \delta$.*

Differential privacy is stronger than leave-one-out-unfairness, as any change to the model—even if it does not actually affect prediction of any point in the distribution—can potentially leak information, and is therefore a violation of differential privacy. This makes sense in the context of privacy, as it concerns an adversarial setting where an attacker is free to interact with a model as-needed to extract information. The focus of fairness is how people receiving an outcome from a model are treated, and thus leave-one-out unfairness focuses on the model's behavior on the data distribution, drawing attention to how changes in the model could affect those who are its likely subjects.

Leave-one-out unfairness does not require randomization in the model's learning rule, whereas differential privacy does. Figure 3 shows an intuitive example of this, where the deterministic learning rule may yield models with unstable outcomes, but only on points

with vanishing probability; for points with non-zero probability, the model's predictions will remain consistent across unit changes to the training data. Moreover, because Definition 1 depends on $\mathcal{D}$, a learning rule may have little leave-one-out unfairness on some distributions, and more on others. However, as Proposition 3.2 shows, differential privacy implies bounded LUF. A proof can be found in the supplementary material.

Proposition 3.3. *Let h be an (ϵ,δ)-differentially private learning rule, and $x \sim \mathcal{D}$ be a point. Then* $\text{LUF}(h,S,x) \leq e^{\epsilon} - 1 + \delta$.

Individual Fairness. Individual Fairness is a Lipschitz condition that aims to formalize the maxim: "similar people ought to be treated similarly". Importantly, in the context of supervised learning this is typically construed as a constraint on *models* rather than learning rules. This stands in contrast to Definitions 2 and 3, which impose a constraint on the latter. Additionally, our definitions do not relate the treatment of individuals to others, but instead measure the degree to which one's treatment by the model may be arbitrarily decided by the composition of the training data. While there is no reason that individual fairness and leave-one-out fairness cannot coincide, there is no a priori reason to believe that they will. In Section 5, we present experimental results on models trained with random smoothing, which has been shown to guarantee individual fairness [61]; shedding further light on the relationship between these two fairness concepts.

We note that leave-one-out unfairness is also related to the definition of memorization introduced by Feldman [24], which we discuss in greater detail in Section 7.

4 LUF IN DEEP MODELS

We characterize the prevalence of leave-one-out unfairness across models trained on several types of data: tabular, time-series, and image data. Importantly, we find that a non-trivial fraction of data (from 3% to 77%) experiences LUF, and moreover, that the prevalence does not appear to depend on model generalization, test accuracy, or dataset size.

Datasets. We perform all of our experiments over five datasets: *UCI German Credit* [17], *Adult* [17], *Seizure* [17], *Fashion MNIST* [59], and Labeled Faces in the Wild [30]. The German Credit data set consists of individuals' financial data, with a binary response indicating their creditworthiness. The Adult dataset consists of a subset of publicly-available US Census data, with a binary response indicating annual income of > 50k. The Seizure dataset comprises time-series EEG recordings for 500 individuals, with a binary response indicating the occurrence of a seizure. Fashion MNIST contains images of clothing items, with a multilabel response of 10 classes. Labeled Faces in the wild consists of unconstrained pictures of individuals' faces, with labels connoting the identity of the individual in each picture. Further information about these datasets and the preprocessing steps we apply can be found in the supplementary material. Table 2 contains the accuracy and generalization error for each baseline model h_S for all datasets.

Setup. For all experiments, we train models using Keras 2.4.3 with TensorFlow 2.0. In keeping with common practice, we set the random seeds used by Python, numpy, and Tensorflow. Beyond this, in order to isolate the effect of leave-one-out unfairness from other sources of instability, we use the same random initialization of model parameters across models in the same experiment, and we turn off non-determinism in GPU operations [55]. This effectively makes the learning rule h deterministic, so that when measuring LUF, the probabilities in Definition 2 are $\in \{0,1\}$. We note that, in the case of, LFW, an additional source of instability remains in the process that produces pairs of faces dynamically during training. This is necessary in order for the model to encounter a sufficiently high number of face pairs during training while being bound to memory constraints. We provide results of the same experiments over a smaller, static dataset in the supplementary material, with similar LUF behavior but lower accuracy.

As it would be prohibitively expensive to train $|S|$ models for the datasets S listed above, we instead measure differences over a fixed number of training sets obtained by randomly deriving from each dataset: a training set S, a set $O \subseteq S$ of size 100 that consists of points drawn randomly from test data (i.e. with which to create 100 different $S^{(\setminus i)}$), and a test set. We train a "baseline" deep model h_S with which to calculate the differences in prediction resulting from removing a point from O from S. For each $z_i \in O$, we train $h_{S^{(\setminus i)}}$ by removing z_i from S. For each $h_{S^{(\setminus i)}}$, we estimate $\text{LUF}(h,S,x)$ for all x in the dataset by measuring the differences between $h_S(x)$ and $h_{S^{(\setminus i)}}(x)$, and taking the maximum difference over the sample of 100 leave-one-out points O. Since the distribution that each training set S comes from is a uniform distribution over the entire dataset, this is measuring $\mathbb{E}_x[\text{LUF}(h,S,x)]$ for each training set S and learning rule h. A step-by-step explanation of this calculation is given in the supplementary material. Due to the cost, for LFW we train 50 $h_{S^{(\setminus i)}}$ models, i.e., in this case we set $|O| = 50$.

To verify that the leave-one-out unfairness is a property of the models and not an unavoidable consequence of training a machine learning model on the presented datasets, we also train linear models on the same datasets with the same method, and compare the leave-one-out unfairness of these linear models to their deep counterparts.

The majority of our results displaying the extent of expected LUF in deep models center around the use of one architecture, seed, and set of hyper-parameters per dataset, in order to keep as many variables controlled as possible. To ensure that the behavior described is consistent, we present experiments displaying the effect of changing architecture and random seed on our main results in Figure 5. The main set of models for German Credit and Seizure datasets have three hidden layers, of size 128, 64, and 16. Models on the Adult dataset have one hidden layer of 200 neurons. The FMNIST model is a modified LeNet architecture [36]. This model is trained with dropout. The LFW face-matching model consists of a concatenation layer composing the two input images, a 4-layer convolutional stack, followed by a dense layer, and a Sigmoid output. German Credit, Adult, and Seizure models are trained for 100 epochs; FMNIST and LFW models are trained for 50. German Credit models are trained with a batch size of 32, FMNIST 64, and Adult, Seizure, and LFW used batch sizes of 128. German Credit, Adult, Seizure and LFW models were trained with Adam ($lr = 1.e^{-3}$), and FMNIST with SGD ($lr = 0.1$).

The experiments outlined above were also performed on models with two other architectures per dataset, in order to compare results across architecture, presented in Figure 5. For German Credit and Seizure datasets, one additional architecture was a shallower model of a 1-hidden layer model of size 100, and the other a narrower model

	Deep		PGD		Trades		Smoothed		Linear	
dataset	*base acc*	*gen err*	*base acc*	*gen err*	*base acc*	*gen err*	*base acc*	*gen err*	*base acc*	*gen err*
German Credit	0.7500	0.2500	0.7400	0.22	0.745	0.253	0.755	0.245	0.745	0.0175
Adult	0.8418	0.0344	0.8226	-0.0019	0.83217	0.0845	0.8390	0.0180	0.8400	0.000
Seizure	0.9736	0.0264	0.9770	0.000	0.9672	0.0083	0.9754	0.0246	0.8113	0.0043
FMNIST	0.9111	0.0211	0.7876	0.0099	0.9016	0.0700	0.8678	0.0269	0.8368	0.0145
LFW	0.8695	0.0597	-	-	-	-	-	-	0.5790	-0.0755

Table 2: Test accuracy and generalization error for all h_S models.

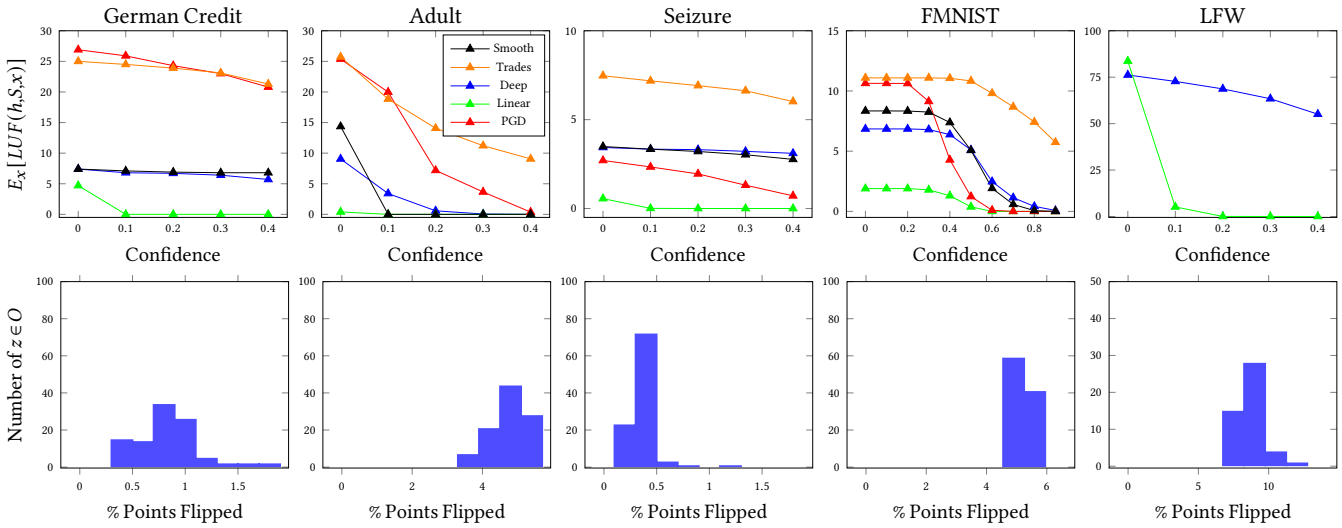

Figure 4: *Top row:* Prediction confidence on the horizontal axis, percentage of stable points experiencing LUF (i.e., $E_x[LUF(h,s,x)]$) on the vertical axis. For FMNIST, confidence is calculated as the absolute difference between the two most confidently predicted classes; for other datasets, confidence is $|h_S(x) - 0.5|$. Note the differences in scale between the graphs; adversarial German Credit and Adult models display especially high leave-one-out unfairness, as well as LFW. *Bottom Row*: A bar chart displaying what percentage of points in the dataset are affected by *each one* of the points taken out. Each bar shows the number of points in O (left-out points) whose absence changed the prediction of the percentage of points shown on the x axis. Notably, every single point that was taken out of the dataset affected at least one other individual's prediction. Note the difference in scale on the x axis.

of 3 hidden layers of sized 64, 32, and 8. For the Adult dataset, the additional models were a narrower 1-hidden layer of size 100, and a deep model with the same architecture as the main German Credit models. For FMNIST, we trained a shallower model with one set of layers removed, as well as a model with no dropout. Finally, for LFW, we compare with a ResNet50 [28] model, pre-trained on ImageNet, and modified to take in two inputs and have a Sigmoid output, as well as a model whose filters are twice the size of the original model. For experiments comparing the extent of expected LUF across models seeded differently, we perform the main experiments outlined in the paragraphs above over 5 different random seeds for all tabular and time series datasets, and three different random seeds for image datasets. Further details on model construction can be found in the appendix.

LUF in Deep Models. Figure 4 shows the prevalence of leave-one-out unfairness on all five datasets. The first row plots the percentage of individuals x experiencing LUF(h,S,x): i.e., $E_x[LUF(h,S,x)]$, ranging over the confidence of the baseline model's prediction. On every dataset examined, deep models display nontrivial expected LUF, ranging from ~4% to ~77%. The second row shows the number of points in $z_i \in O$ (out of 100) that lead to a given percentage of individuals x having their predictions changed when only z is removed from the dataset. The percentage per point on the X axis, and the number of points that change this percentage of outcomes is on the Y axis. Notably, the removal of each point sampled lead to an $h_{S^{\setminus i}}$ model that changed the predictions of at least one other point, suggesting that leave-one-out unfairness is in fact very common.

The results show that leave-one-out unfairness cannot be reliably predicted given test accuracy, and more notably, generalization error (shown in Table 2). While it may seem natural that models with higher accuracies display less LUF, the deep model on the Adult dataset has an accuracy ~10% higher than the German Credit dataset, yet the German Credit dataset has approximately 2% fewer individuals experiencing LUF. Even more impressively, the LFW model has higher accuracy than both German Credit and Adult models, by 12%

and 2% respectively, yet has a much higher expected LUF of ~77%, compared to 7% and 10%. Similarly, following intuitions from model stability, lower generalization error may naturally seem to coincide with lower levels of LUF. However, the German Credit model has a generalization error of ~25%, yet has lower LUF than both the Adult model, with generalization error of just ~3%, and the LFW model, with generalization error of ~5%. Indeed, while these results will be further discussed in the next section, it is worthy of note that the PGD model on the Adult dataset has essentially zero generalization error, yet has a very high percentage of individuals experiencing leave-one-out unfairness (~25%), while the deep model on the Adult dataset has generalization error of ~3.5% and has around 10% of individuals experiencing LUF. While we did not explicitly control for accuracy or generalization error, these results are evidence that LUF does not depend on these metrics.

Also of note is that LUF does not decrease with dataset size—FMNIST and German Credit are the largest and smallest datasets, with training set sizes of 60,000 and 800 respectively, yet FMNIST displayed similar LUF to German Credit (within 1%). The Adult dataset is also larger than German Credit (~$|S| = 15{,}000$) and displays more expected LUF.

Perhaps most importantly, confidently-predicted points are not immune from leave-one-out unfairness in deep models: on the majority of the datasets, a substantial portion of points with high LUF were predicted with confidence greater than 0.9 by the baseline model. This is illustrated by the fact that the curves displaying the number of points versus baseline model confidence do not drop off sharply in all models except for those on the Adult dataset. This is an interesting manifestation of miscalibration in deep models: some confident decisions may still be somewhat arbitrary, in that they are sensitive to the specific makeup of the training set.

Consistency Under Varying Conditions. We provide calculations of expected LUF over all datasets in deep models where the architecture and random seed differ, in order to ensure that the results are consistent across different modeling choices.

The results are presented in Figure 5. While there is some variation in expected LUF, no modeling choice explored eradicates the behavior. Interestingly, certain architectures seem to exacerbate or diminish LUF: a deeper model increases LUF in the Adult dataset by nearly 10%, and removing dropout from the FMNIST model, as well as increasing the filter size on LFW, have a similar effect. This may warrant further study to find potential mitigation techniques through architecture selection, however, no pattern is immediately noticeable: for example, while a shallower model exhibited lower expected LUF on the German Credit dataset than the baseline model, the same shallow architecture exhibited more expected LUF than the baseline on the Seizure dataset, which shares the same architecture as the German Credit baseline model. Random seed also affects the prevalence of expected LUF, to a slightly lesser extent for all models but LFW. Broadly, however, the results show that LUF is not an artifact of any one particular set of training conditions.

Linear Models. We also provide the results for the same experiments on linear models to calibrate against a more stable learning rule that yields less complex models: observe the green line in Figure 4. These results show that LUF is not inherent to the data. While there are points that are treated leave-one-out unfairly, they are substantially fewer—with the exception of LFW, where the learning task is markedly more complex than the other datasets, and unsuitable for a linear model. Additionally, the overwhelming majority of points treated leave-one-out unfairly in linear models are not confidently predicted—in fact, in all models but FMNIST, there are no points treated leave-one-out unfairly that are predicted with a difference of more than 10% from 50% confidence.

This result agrees with intuition—linear boundaries are smooth, and linear regression is stable. If the introduction of a point does shift the boundary, it is likely that only points already close to the decision boundary (i.e., low-confidence points) are affected. Deep models can have arbitrarily complex decision boundaries, which appears to be closely-related to LUF. As the phenomenon of memorization [24, 63] suggests, and these results support, deep models have the capacity to "overreact" to the presence of individual entries in their training data. Figure 1 illustrates this further in a low-dimensional setting. Not only can the region around the left-out point potentially change, but there are may also be far-reaching effects on the decision boundary beyond the neighborhood of the left-out point. These changes will affect not just the predicted label of new points, but also their assigned confidence score. While intuitions that are valid in low-dimensional settings do not always transfer to high dimension, this may nonetheless provide some intuition behind the factors that contribute to leave-one-out unfairness.

5 LUF AND ROBUST CLASSIFICATION

Calls to mitigate adversarial examples [45, 54] have motivated a significant amount of research aimed at producing robust classifiers [14, 40, 58]. Recent results have shown that some of these techniques can even be repurposed to ensure individual fairness [61], and moreover, that they often produce deep models that admit more interpretable feature attributions [22, 31, 43]. Intuitively, these findings could suggest that robust prediction methods rely on "robust features" [31] that align more closely with human understanding of the problem domain, and whose presence in the model may be accordingly less dependent on individual points in the training data.

In this section, we explore this conjecture by measuring the incidence of leave-one-out unfairness with two robust classification methods: adversarial training, and randomized smoothing. We find that models trained adversarially using projected gradient descent (PGD) [40] as well as models trained with the TRADES algorithm [64] have significantly higher rates of LUF, in most cases approximately doubling the number of unstable points over standard training. On the other hand, models that are made robust by post-hoc smoothing with Gaussian noise [14] almost always have similar rates of expected LUF. Taken together, these results suggest that LUF and robustness are not inherently tied to one another, but that certain classes of models may provide beneficial properties for both, warranting further study.

Setup. We use the same experimental setup as in Section 4 for measuring leave-one-out unfairness. In these experiments, we only train deep models. For adversarial training, we use PGD with an ℓ_2 radius $\epsilon = 3.0$ and 10 PGD steps on FMNIST and Seizure datasets. For the Adult and German Credit datasets, we use radius $\epsilon = 1.0$. On the German Credit dataset, we use the ℓ_∞ norm. The radius remained

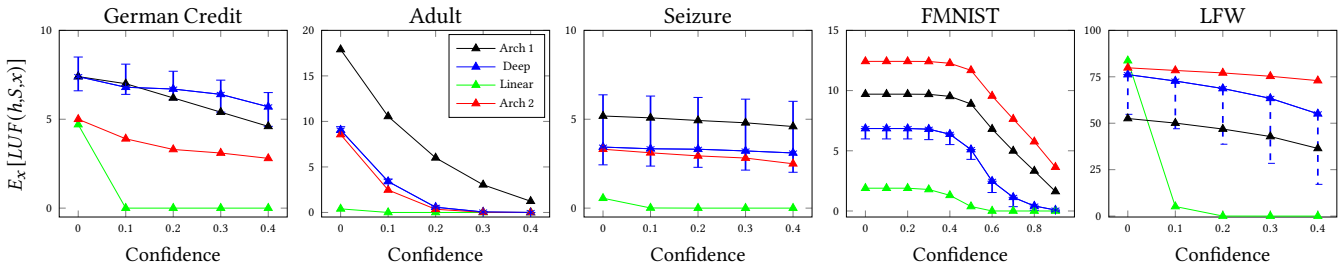

Figure 5: Effect of random seed and architecture on LUF results in deep models from Figure 4. The red and green plots show LUF for models of slightly different architecture, as described in the experimental setup, and the bars on the blue line show the minimum and maximum LUF values over 5 random seeds on the main architecture shown in main results. Notice the difference in scale across the graphs.

the same between PGD and TRADES training. We determined the radius for adversarial training by finding the minimum distance (with respect to the adversarial norm) between any two points of different classes over a large sample of the dataset. If this was impossible because this distance was zero, we chose a distance smaller than that between over 99% of cross-class pairs of points in the sample. For TRADES training, we used all of the same hyperparameters as PGD training, with the addition of the TRADES parameter, which was 1 for Adult and German Credit, and 10 for Seizure and FMNIST. Notice that, for face-matching problems, the threat model for finding adversarial examples is less clear—e.g., it is not obvious if the attacker has access to individual images, or pairs of images. As we are unaware of an established threat model for face-matching, we do not evaluate LFW in this section. For randomized smoothing, we take 1,000 Gaussian samples with $\sigma^2 = 0.1$ for the Adult and Seizure datasets, 10,000 samples with $\sigma^2 = 0.05$ for FMNIST, and 2,000 samples with $\sigma^2 = 0.05$ for German Credit. While Cohen et al. [14] report needing more smoothing samples to achieve strong adversarial guarantees, our goal in these experiments is to measure LUF, which we found to be insensitive to additional samples beyond the numbers reported above. The accuracy of these models is shown in Table 2.

Results and Discussion. The results are shown in Figure 4. The most immediate trend is the degree to which PGD and TRADES adversarial training worsens LUF: approximately by a factor of two across all datasets, and by a factor of nearly three on the German Credit dataset. Seizure is a partial exception in that the PGD training does not worsen LUF, but TRADES training does. While adversarial training produces models that are more invariant to small changes in their inputs, these results show that the training procedure itself can be unstable. This may be related to prior work demonstrating that adversarially-trained models are more vulnerable to *membership inference* [53, 62], a privacy attack that exploits memorization to leak information about training data. While membership vulnerability does not necessarily imply greater LUF, these experiments show that in many cases the two phenomena may be related. We also note that these results do not necessarily contradict the "robust feature" hypothesis proposed by Ilyas et al. [31], as robust learned features need not generalize across large portions of the dataset.

Turning to the curves labeled "Smooth" in Figure 4, it is clear that randomized smoothing leads to qualitatively different leave-one-out unfairness results. On most datasets, smoothing had little effect ($< 1\%$ difference) on expected LUF. Beyond suggesting that leave-one-out unfairness is independent of robustness, these results also point to the fact that individual fairness and LUF are related, but separate notions. Randomized smoothing guarantees individual fairness for weighted ℓ_p metrics [61], but has a negligible effect on leave-one-out unfairness.

Looking at the geometry of these models can shed further light on the differences in results between PGD training and randomized smoothing. As suggested by Figure 1, deep model decision boundaries have the potential to be very sensitive to individual points, and this sensitivity may affect regions of the decision boundary far beyond the local neighborhood of the point in question. This could contribute to leave-one-out unfairness, as the predictions of points in regions shifted by a training points' addition or removal will change. Adversarial training may in some cases intensify the boundaries' sensitivity to training points by penalizing inconsistent predictions in any direction within ϵ away.

Alternatively, a smoothed model returns the expected prediction over a continuous distribution centered at each point, rather than the value of the underlying model at only one point. While this does not remedy larger boundary changes stemming from instability, it likely does not exacerbate them, as evidenced by the effects in Figure 4.

6 DISCUSSION

Our study focused on instability to changes in training data, as this type of stability is particularly well-studied due to its relevance to generalization and privacy. However, there are other potential sources of instability that may lead to arbitrary outcomes as well: for example, random initialization, batching order, and model architecture. If a difference in any of these choices results in a difference in outcome for an individual—e.g., if a change in random initialization frequently leads to a change in predicted credit risk for someone—then this too could be seen as unfair, as it would call into question the robustness of any supposed justification.

To establish a preliminary understanding of the degree to which these sources introduce changes in outcome similar to LUF, we experimentally investigate the percentage of changed outcomes resulting from varying the random seed prior to initializing and training models, as well as from the choice of model architecture. Figure 6 shows these results for all of the datasets studied in Section 4, alongside the corresponding measurements for LUF. The experimental setup largely follows that described in Section 4. We isolate the effect of each potential variable causing instability unfairness (architecture,

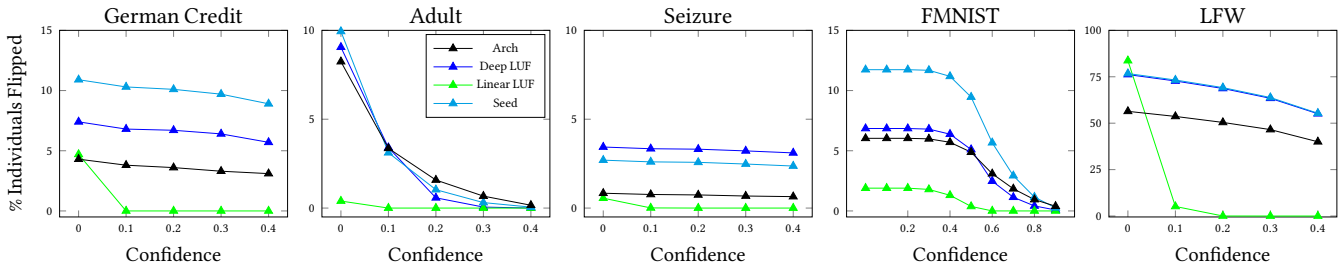

Figure 6: Arbitrariness in decision outcome as a result of changes in random seed, and small changes in architecture, are presented alongside expected LUF, i.e. arbitrariness from small changes in the training set. Calculation methods are described in 6. We present these results to motivate a wider connection between learning algorithm stability and fairness, beyond LUF. Notice the difference in scale across graphs.

random seed, and leave-one-out unfairness) in its own experiment; keeping other sources of instability controlled. For the random seed experiments, we train the same model with 100 different random seeds and calculated the effects of instability in the same manner as calculating LUF described in Section 4; for the experiments calculating the fairness effects of changes in architecture, we train the model on three different architectures, as described for the experiments verifying consistency in LUF in Section 4. Further information on the architectures considered can be found in the supplementary material.

As Figure 6 shows, any of these aspects in a model can affect model behavior over a substantial percentage of the overall dataset. Interestingly, LUF seems to have a more consistent effect across points with high prediction confidence than arbitrariness resulting from a change in architecture. LUF seems to have a similar effect to changing random seed and initialization, as changing seed produces a larger effect in FMNIST and German Credit, but a smaller effect in Adult and Seizure models. While these other sources of instability unfairness are interesting avenues for future work, we focus on leave-one-out unfairness in this paper due to its useful connections to other areas of the machine learning literature, bridging the fields of fairness to those of stability and privacy as discussed in Section 3, and also to the field of robustness, as explored in Section 5.

7 RELATED WORK

Leave-one-out unfairness views the problem of learning instability [11, 12] from a fairness perspective. While deep learning is generally understood not to enjoy strong stability properties, our results are among the few systematic studies of the extent, and potential ramifications, of their instability. Hardt et al. show that even non-convex models trained using Stochastic Gradient Descent remain stable over a small number of iterations, and that popular heuristics like dropout and ℓ_2 regularization help [27], and provide some experimental demonstrations. Towards achieving stability in deep learning, Kuzborskij et al. [35], develop a screening protocol for choosing random initalizations that improve stability.

Memorization, as defined by Feldman [24], is a symptom of model instability where a model predicts the correct output on a given point if it is in the training set, and incorrectly otherwise. There has been much recent work unearthing the potential for memorization in deep neural networks [63], discussion about the extent of the phenomenon in practice [6] as well as arguments for its usefulness [24]. Memorization is closely related to leave-one-out unfairness in it is a measure of stability, and crucially, focuses on how instability affects a given point, rather than an average. However, leave-one-out fairness is much broader than memorization. Memorization quantifies how much removing a given point from the training set affects that whether that particular point is predicted correctly. Leave-one-out fairness quantifies how the consistency, not the error, of a given point's prediction is affected by *any other point*.

A well-known meeting point of stability and privacy is differential privacy [18], which quantifies privacy risk in terms of a uniform, information-theoretic notion of stability. Leave-one-out fairness is related to, but weaker than, differential privacy, as shown in Section 3. Instability also worsens concrete privacy attacks: oversensitivity to the training set can affect a model's parameters, which can be leveraged to perform membership inference [37, 52, 60]. Our experiments in Section 5 may suggest that this phenomenon has a connection to leave-one-out unfairness, in that adversarial training increases both LUF and the potential for membership inference attacks [53, 62].

There is little work that connects *fairness* and stability. Leave-one-out fairness is an individual-based fairness notion. While there are several definitions of "individualized" fairness [19, 20, 32, 34], they are rarely operationalized in common fairness testing platforms, as they can be difficult to calculate. In addition to already-noted differences from prior notions of fairness, expected LUF can be effectively measured on real datasets to give insight into whether an individual may be subject to unfair treatment at inference time.

8 CONCLUSION

We present *leave-one-out* fairness, a connection between algorithmic stability and fairness. We demonstrate the extent to which deep models are leave-one-out unfair, and experimentally showed that this behavior does not depend on generalization error. Interestingly, adversarial training worsens leave-one-out unfairness in deep models, while random smoothing often mildly mitigates it, showing that leave-one-out fairness is not dependent on robustness or individual fairness. These results may suggest an interesting geometric intuition of deep networks' sensitivity to their training points. Finally, we note that LUF may be undesirable in sensitive applications, as it casts doubt on the justifiability of a model's decision.

Acknowledgments

This paper is based on work supported by the National Science Foundation under Grants No. CNS-1943016 and CNS-1704845.

REFERENCES

[1] *Henry v. United States*, volume 361 U.S. 98. 1959.
[2] Equal credit opportunity act (regulation b). https://www.fdic.gov/regulations/laws/rules/6500-200.html, 2011.
[3] European parliament and council of european union (2016) regulation (eu) 2016/679. https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679&from=EN, 2016.
[4] Alejandro Acien, Aythami Morales, Ruben Vera-Rodriguez, Ivan Bartolome, and Julian Fierrez. Measuring the gender and ethnicity bias in deep models for face recognition. In *Iberoamerican Congress on Pattern Recognition*. Springer, 2018.
[5] Peter Addo, Dominique Guegan, and Bertrand Hassani. Credit risk analysis using machine and deep learning models. *Risks*, Apr 2018.
[6] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2017.
[7] Dmitrii Babaev et al. Et-rnn: Applying deep learning to credit loan applications. In *KDD*, 2019.
[8] Mihalj Bakator and Dragica Radosav. Deep learning and medical diagnosis: A review of literature. *Multimodal Technologies and Interaction*, Aug 2018.
[9] Ramnath Balasubramanian et al. Insurance 2030: The impact of ai on the future of insurance. *McKinsey & Company*, 2018.
[10] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, 2016.
[11] J Frédéric Bonnans and Alexander Shapiro. *Perturbation analysis of optimization problems*. Springer Science & Business Media, 2013.
[12] Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of machine learning research*, 2(Mar):499–526, 2002.
[13] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, 2018.
[14] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.
[15] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, 2016.
[16] Anupam Datta, Matt Fredrikson, Gihyuk Ko, Piotr Mardziel, and Shayak Sen. Use privacy in data-driven systems: Theory and experiments with machine learnt programs. In *ACM SIGSAC Conference on Computer and Communications Security*, 2017.
[17] Dheeru Dua and Efi Karra Taniskidou. UCI machine learning repository. https:/ive.ics.uci.edu/ml, 2017.
[18] Cynthia Dwork. Differential privacy, 2006.
[19] Cynthia Dwork and Christina Ilvento. Fairness under composition. *CoRR*, abs/1806.06122, 2018.
[20] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Innovations in Theoretical Computer Science*, 2012.
[21] Geert Litjens et. al. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 2017.
[22] Christian Etmann, Sebastian Lunz, Peter Maass, and Carola-Bibiane Schönlieb. On the connection between adversarial robustness and saliency map interpretability. In *ICML*, 2019.
[23] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.
[24] Vitaly Feldman. Does learning require memorization? A short tale about a long tail. *CoRR*, abs/1906.05271, 2019.
[25] Clare Garvie, Alvaro Bedoya, and Jonathan Frankle. The perpetual lineup, 2016.
[26] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, 2016.
[27] Moritz Hardt, Benjamin Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, ICML'16, 2016.
[28] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
[29] Kashmir Hill. Wrongfully accused by an algorithm. *The New York Times, June*, 24, 2020.
[30] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. 2008.
[31] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems 32*. 2019.
[32] Matthew Joseph, Michael Kearns, Jamie H Morgenstern, and Aaron Roth. Fairness in learning: Classic and contextual bandits. In *Advances in Neural Information Processing Systems*, 2016.
[33] Margot E Kaminski. The right to explanation, explained. *Berkeley Tech. LJ*, 34: 189, 2019.
[34] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. In *Advances in Neural Information Processing Systems*, 2017.
[35] Ilja Kuzborskij and Christoph H. Lampert. Data-dependent stability of stochastic gradient descent. In *ICML*, 2018.
[36] Yann LeCun, LD Jackel, Léon Bottou, Corinna Cortes, John S Denker, Harris Drucker, Isabelle Guyon, Urs A Muller, Eduard Sackinger, Patrice Simard, et al. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural networks: the statistical mechanics perspective*, 261:276, 1995.
[37] Klas Leino and Matt Fredrikson. Stolen memories: Leveraging model memorization for calibrated white-box membership inference. 2020.
[38] Zachary C Lipton. The mythos of model interpretability. *Queue*, 16(3):31–57, 2018.
[39] Elizabeth Lopatto. Clearview ai ceo says 'over 2,400 police agencies' are using its facial recognition software, 2020.
[40] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.
[41] Ilya Mironov. Rényi differential privacy. In *Proceedings of 30th IEEE Computer Security Foundations Symposium (CSF)*, 2017.
[42] Madhumita Murgia. Who's using your face? the ugly truth about facial recognition|. *Financial Times*, 2019.
[43] Adam Noack, Isaac Ahern, Dejing Dou, and Boyang Li. Does interpretability of neural networks imply adversarial robustness? *CoRR*, abs/1912.03430, 2019.
[44] Orestis Papakyriakopoulos, Simon Hegelich, Juan Carlos Medina Serrano, and Fabienne Marco. Bias in word embeddings. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020.
[45] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS P)*, 2016.
[46] Manish Raghavan, Solon Barocas, Jon Kleinberg, and Karen Levy. Mitigating bias in algorithmic hiring: Evaluating claims and practices. FAT* '20, New York, NY, USA, 2020. Association for Computing Machinery.
[47] J Schuppe. How facial recognition became a routine policing tool in america, 2019.
[48] Andrew D Selbst and Julia Powles. Meaningful information and the right to explanation. *International Data Privacy Law*, 7(4):233–242, 12 2017. ISSN 2044-3994.
[49] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, 11, 2010.
[50] Aditya Shinde et al. Comparative study of regression models and deep learning models for insurance cost prediction. In *ISDA*, 2018.
[51] Justin Sirignano, Apaar Sadhwani, and Kay Giesecke. Deep learning for mortgage risk. *CoRR*, abs/1607.02470, 2016.
[52] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017.
[53] Liwei Song, Reza Shokri, and Prateek Mittal. Membership inference attacks against adversarially robust deep learning models. In *2019 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2019.
[54] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2013.
[55] tensorflow-determinism Python package. Available at: https://pypi.org/project/tensorflow-determinism/. Retrieved on 6/5/2020.
[56] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019.
[57] James Vincent. Nypd used facial recognition to track down black lives matter activist. *The Verge, August*, 2020.
[58] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.
[59] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.
[60] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, 2018.
[61] Samuel Yeom and Matt Fredrikson. Individual fairness revisited: Transferring techniques from adversarial robustness. In *IJCAI*, 2020.
[62] Samuel Yeom, Irene Giacomelli, Alan Menaged, Matt Fredrikson, and Somesh Jha. Overfitting, robustness, and malicious algorithms: A study of potential causes of privacy risk in machine learning. *J. Comput. Secur.*, 28(1), 2020.
[63] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *CoRR*, abs/1611.03530, 2016.
[64] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, 2019.