Attacking Machine Learning Models for Social Good *

Vibha Belavadi[1], Yan Zhou[1], Murat Kantarcioglu[1], and Bhavani Thuriasingham[1]

University of Texas at Dallas, Richardson TX 75080, USA

Abstract. As machine learning (ML) techniques are becoming widely used, awareness of the harmful effect of automation is growing. Especially, in problem domains where critical decisions are made, machine learning-based applications may raise ethical issues with respect to fairness and privacy. Existing research on fairness and privacy in the ML community mainly focuses on providing remedies during the ML model training phase. Unfortunately, such remedies may not be voluntarily adopted by the industry that is concerned about the profits. In this paper, we propose to apply, from the user's end, a fair and legitimate technique to "game" the ML system to ameliorate its social accountability issues. We show that although adversarial attacks can be exploited to tamper with ML systems, they can also be used for social good. We demonstrate the effectiveness of our proposed technique on real world image and credit data.

Keywords: Adversarial Machine Learning · Adversarial Attacks · Artificial Intelligence Fairness · Data Privacy .

1 Introduction

Increasingly, machine learning (ML) models have been deployed in many critical applications ranging from credit scoring to triaging patients for emergency care (e.g., [19]). Unfortunately, using ML models for critical decision-making tasks can raise fairness and privacy concerns. For example, an ML model used to predict criminal recidivism has been shown to be biased against a certain subgroup [25]. In other cases, ML models could be used to predict some sensitive information. For instance, it has been shown that ML models could predict sexual orientation based on Facebook likes and/or profile images [30]. The sexual orientation information by itself may be sensitive and even the existence of an accurate ML model could result in significant privacy loss.

To address some of these issues, there is an active ongoing research on fairness and privacy in ML. The proposed techniques range from new algorithms that produce fair ML models (see the survey for more details [8]) to differentially

* The research reported herein was supported in part by NIH award 1R01HG006844, NSF awards CNS-1837627, OAC-1828467, IIS-1939728 and ARO award W911NF-17-1-0356

private machine learning models that protect individual privacy (see the survey for more details [15]). Unfortunately, most of these techniques require the buy-in of the organization that is deploying the ML model and may not be easily leveraged by end users in the already deployed ML models.

Although, some existing privacy regulations such as GDPR [9], if requested, require ML based decisions to be audited by humans. Still, as the recent research indicates, it is not always possible for humans to detect potential biases in the ML models (e.g., [2]) even if the ML decisions are explained using explainable AI techniques.

In this work, we propose a complementary approach that tries to protect individual privacy and increase fairness by "attacking" the ML model directly. In other words, the user may modify some of his/her data, the input to the ML model, so that the privacy sensitive decisions that could be generated by the ML model are impacted and the potential bias of the ML model is reduced. Our approach is based on the observation that many of the ML models are not robust against adversarial attacks that modify inputs to the ML models (e.g., adding background noise to deceive an image classifier). Therefore, such approach can be used to hinder ML models that try to predict sensitive information and increase fairness by changing "biased" decisions without any cooperation from the organizations that deploy the ML models.

Compared to traditional adversarial machine learning settings, in this context, we want to make sure that our attacks are ethical and legal. In other words, it may be illegal to lie about your income in a credit card application but it is acceptable to get a free checking account from a bank to improve your credit score. To address this challenge, we carefully define the *cost of data modification* in the developed "adversarial" attacks so that illegal, unethical, and unfeasible modifications are not considered during the "attack".

The main contributions of this paper could be summarized as follows:

- We provide a framework that improves privacy and fairness without the cooperation of the ML model owners.
- Our framework is carefully designed by specifying appropriate cost functions to only consider data modifications that are legal and ethical.
- We empirically show the utility of this framework in two different applications (image classification and credit application).

The rest of the paper is organized as follows: in Section 2, we discuss the related work. In Section 3, we provide a generic framework that shows how to deploy adversarial attacks for improving privacy and fairness and show the initiation of this generic model in two application domains. In section 4, we show the utility of the proposed framework in two different applications via extensive empirical evaluation. Finally, in section 5, we conclude with the discussion of our results and the future work.

2 Related Work

Adversarial attacks have become a major threat to applications that heavily rely on the integrity and accuracy of machine learning models. Adversarial learning has been an active research area for years [10, 21, 17, 35, 4, 34], but only catches more awareness as the deep learning technique becomes popular. Recent studies on adversarial attacks mainly target gradient-based attacks against deep neural networks for image classification [28, 13, 23, 5, 6].

More recently, concerns on adversarial attacks are being raised in other machine learning application domains such as finance and health care where modifying data is more restricted by data domain constraints [14, 24]. Ballet et al. [3] demonstrate how adversarial samples can be crafted for tabular data in the finance domain. They discuss the unique challenge specific to models trained on tabular data: how to make the modified sample, such as a loan application, remain credible and relevant for a potential expert eye? Unlike image data, tabular features are not interchangeable and less readable. For people with expert knowledge, only a small subset of features is most critical when making decisions. Therefore, adversarial attacks should avoid this subset of important features when modifying samples. An empirical study on tabular data attacks and their detection and mitigation by model interpretation and reducing attack vector size has been presented in [16].

The influence of adversarial attacks has also been investigated in the context where users can game machine learning systems to gain or protect for better social, economic, moral, and political advantages [24, 18]. For example, Protective Optimization Technologies (POTs) provide the users of machine learning systems with tools to counter or contest the biases and discriminatory harms caused by these systems [18]. For dishonest users gaming the system to gain advantages, such as the approval of a loan application, features critical to the final decisions can be identified and verified to mitigate this kind of attack against the decision-making systems [24].

The threat of adversarial attacks in the applications of computer vision, ranging from self-driving cars to surveillance and security, has become a heated topic recently. A detailed survey can be found in [1]. For the purpose of poisoning attacks, backdoors and patches—digital patterns and their physical realizations deliberately inserted into images to cause misclassification—have been heavily studied in image classification [12, 22, 31].

Deep Learning has become the backbone of various face recognition systems offered by Amazon, IBM, Google, Microsoft and other companies like FacePlus-Plus. Wang & Kosinski [30] applied Deep Learning to test whether the sexual orientation of a person can be accurately predicted better than a human predicting it. They claim that upon transfer learning with VGGFace, they are able to predict sexual orientation with a better accuracy than the human. In wake of such claims, it becomes imperative to be able to safeguard sensitive attributes identifiable from images from such black box models. One such approach is using adversarial examples for good as done by [27]. They have used DCGAN to generate glasses to fool the state-of-the-art face recognition systems. They have

also proposed a general framework where anyone can train their model with a set of generator and a discriminator to create adversarial examples that can fool any machine recognition systems of choice for face images.

Our work is different from these major lines of research in that our "attacks" are strictly constrained to the set of *feasible instances* to which a user data profile can be legitimately modified to achieve fairness and protect privacy. In addition, we take into account the cost of data modification so that changes made to the data would be most feasible and least expensive to the end user. Our objective is to legitimately "attack" the system to mitigate its inherent biases with the least disruption to both parties that must adhere the terms of the contract.

3 Modeling Socially Good Adversarial Attacks

Given a machine learning model f, an instance (x, y) where x is the feature vector for the instance (e.g., a vector of real numbers representing an image) and y is the class value (e.g., y='Heterosexual'). x can be modified to x' by the user such that

$$\begin{aligned} &\arg\min_{x'} \quad c(x, x') \\ &subject \;\; to \; x' \in F_x, f(x') = t \end{aligned} \tag{1}$$

for a set of feasible instances F_x, cost function c that measures the cost of modifying the original instance x by the user, and the desired target class t.

It is important that the instance x' can only adopt modifications that are ethical. Therefore, for a given context, we want to make sure that the set of possible modification F_x is carefully defined. For example, in the case of image processing, we may want to find x' so that the changes to x can be done by adding "eyeglasses". In other words, we may want to make sure that by putting a pair of eyeglasses to an image, a ML model that predicts sexual orientation can be fooled without significantly changing the overall image.

In other domains, there may be other constraints. For example, for a credit card application, it may be illegal to lie about your income. At the same time, opening a new free checking account may be a totally valid and ethical change, especially if this change improves the chance of getting the credit application approved. Therefore, it will be crucial to define the F_x correctly in different contexts.

In addition, to correctly identify F_x, we need to carefully define the cost function c that guides the modification. For example, in credit card application, reducing the existing debt to income ratio may help with the application but it may not be feasible due to the associated monetary cost.

Finally, the attack target t should be carefully designed. For example, for credit application, the t could be the "approved" status. Below, we discuss how our framework could be applied in two important application domains: image classification and credit application evaluation.

3.1 Ethical and Practical Adversarial Attacks for Image Classification

In the case of image classification, we would like to achieve multiple goals. First, we would like the modifications to be concentrated on only certain parts of the images. For example, we may want the modification to be able to printed on a face covering that is commonly worn during covid-19 pandemic. Alternatively, we may want to consider modifications that can be printed and shown on eyeglasses. Therefore, we require the modification to be part of a set $\mathcal{X}_m$ (i.e., modifications concentrated around the eye of the user). In addition, we would like to make sure that the modification ϵ is bounded appropriately in some norm.

$$\begin{aligned} \arg\min_{\epsilon} \quad & f(x+\epsilon) = t \\ subject \;\; to \; & \|\epsilon\| \leq \delta, \epsilon \in \mathcal{X}_m \end{aligned} \tag{2}$$

3.2 Ethical and Practical Adversarial Attacks for Classification with Discrete Attributes

In many domains such as credit application, many of the attributes could be discrete. In addition, due to legal and ethical concerns, we may want to avoid changing certain attributes. In such settings, for each attribute k that could be legally modified, we define the cost of those feasible modifications via cost matrix $C^k_{i,j}$. For attribute k, keeping A^k_i the same has zero cost (i.e., $C^k_{i,i} = 0$). On the other hand, infeasible modifications would have cost of infinity ∞, and the remaining modifications could be assigned appropriate cost value $C^k_{i,j}$ (i.e., cost of changing attribute A^k_i to A^k_j). For example, if the credit applicant has no cell phone, getting a cell phone could be a costly but a feasible transformation. On the other hand, getting rid of the cell phone subscription may not be feasible. Using these observations, we can rewrite Equation (1) as follows:

$$\begin{aligned} \arg\min_{\bigcup_{k=1}^{K}(\{i_k, j_k\})} \quad & \textstyle\sum_{k=1}^{K} w_k \cdot C^k_{i_k, j_k} \\ subject \;\; to \quad & f(M(x)) = t \end{aligned}$$

where M represent the set of modifications that is applied to each attribute (i.e., $M = \bigcup_{k=1}^{K}(\{i_k, j_k\})$), K is the total number of attributes, and w_k is the relative weight of the attribute.

4 Experiments

In the next two sections 4.1 and 4.2, we present the experimental results on the CelebA dataset and the German Credit dataset that illustrate how our proposed framework can be applied in practice.

4.1 Methodology and Experimentation for CelebA dataset

Dataset creation
We train our image classifier on a subset of data from the public CelebA dataset [20]. CelebA dataset is a large scale face-attribute dataset of 202,599 face images from 10,177 celebrity identities with large pose and background variations. The CelebA dataset is richly annotated with 5 landmark locations and 40 binary attributes like 'Arched Eyebrows', 'Eyeglasses', 'Gender', 'Smiling', 'Wearing Hat' etc. We preprocess our training dataset by first extracting 68 facial landmarks using the dlib features from the target image. If an image has no dlib features: either because the image has no facial landmarks or because the face is too small to be detected, we eliminate the image. We then scale the image for convergence during the training process. We also augment our dataset to consider rotation, random cropping, and horizontal flip variants of the same image. By data augmentation, we intend to artificially increase the data size and thus ensure our target model is generalizable on real data. Figure 1 demonstrates the different image augmentation techniques used. For each original image, we use four augmented images for the training process.

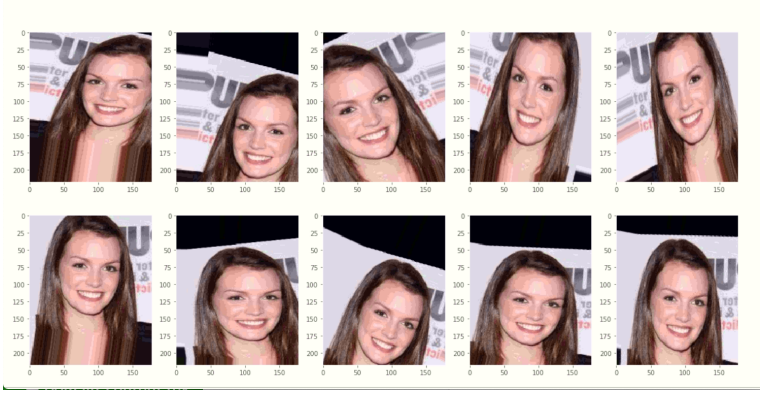

Fig. 1. Data Augmentation used to improve classifier accuracy.

Model training
For our experiments, we have chosen the concept of Gender to train and generate our ethical adversarial examples. [1] We trained our gender model with 20,000 male and 20,000 female examples using transfer learning [33] on the VGGFace model with VGG16 architecture [29]. We chose to transfer learn on the VGGFace model as it has been trained on 2.6 million faces of 2,622 celebrities and hence

[1] Although the gender information is not privacy sensitive, we use this as a substitute for more privacy-sensitive concept such as sexual orientation.

can robustly extract the high level facial features from our images. In our custom model, we first extract the model features from the penultimate layer in our model. We do so by freezing the blocks (specifying their learning rate to be 0). These features are then fine-tuned and further trained by passing them through the final convolution block and the three custom convolution blocks defined on top of it. The final convolution block has relatively smaller learning rate for fine-tuning purposes compared to the custom convolution layers. We train our model using softmax loss. For comparison purposes, we have also trained a gender model on the inception_v3 architecture, though our adversarial attacks will be primarily on our custom VGG16 model. Table 1 presents the training and validation accuracy of the VGG16 and the inception_v3 architectures. As can be observed, the gender concept is successfully learned for the CelebA dataset.

Table 1. Determining gender on the basis of the image

model	f_train_acc	f_val_acc
inception_v3_model	97%	93%
VGG16_model	94.75%	94.44%

Attack mechanism

We use the attack mechanism developed in [26] to attack the gender concept using the artifact of eyeglasses. We first align our data sample to be attacked using target landmarks (canonical pose marks). Once the data is aligned, we choose good candidate images for attack and preprocess them. In our setting, an image is a good candidate image if 1) it is classified correctly without any perturbation and 2) the difference in probability between the correct and incorrect classes is more than 3%. We chose the 3% threshold as we want the classifier to be able to confidently predict the class better than random guessing (50% probability). 3% ensures that the winning probability of the correct class is 51.5% and the other class is 48.5%. We then normalize our images by subtracting the standard normalization constant from them.

To satisfy the constraint based nature of our attack, we perform modifications only on our artifact (eyeglasses) added to the face. We ensure this by limiting perturbation area on the artifact's location on the image. In the case of our artifact, i.e eyeglasses, we focus around the eyes in the specific location of eyeglasses. Before performing the attack, we first initialize our artifact (eyeglasses) to a set of random starting colors to provide an "easy" starting point for perturbation. If any of the starting colors causes change in the original classification, we hold on to that specific initialization for our attack, else we randomly choose one from the set. We show an example of initialization in Figure 2, where we have the eyeglasses artifact before and after initialisation. Given the exact location of our artifact, we selectively normalize the gradients by replacing them with 0 in non-artifact areas of the input and normalizing them with respect to the

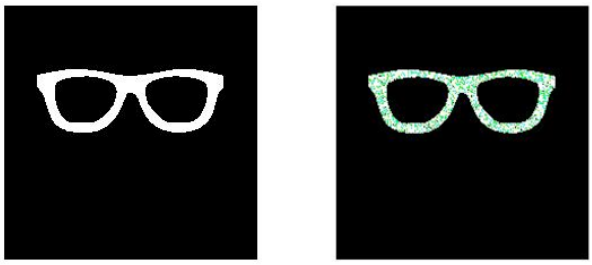

Fig. 2. Random color initialization of our artifact

maximum gradient value otherwise. Once our gradients are normalized, we then perturb them by taking a small step-size in the direction of the gradient. We keep adding the perturbations to the gradient till we flip at least half of the images of the batch. Since we previously initialized the gradients of non-artifact based areas with 0, we guarantee to perturb only the gradients of the artifact region. For this experiment, we chose 279 female candidate images and 266 male candidate images. In both cases, we were able to successfully attack all the chosen images and achieve an attack success rate of 100%. Some of the adversarial examples and their corresponding base images are shown in figures 3 and 4.

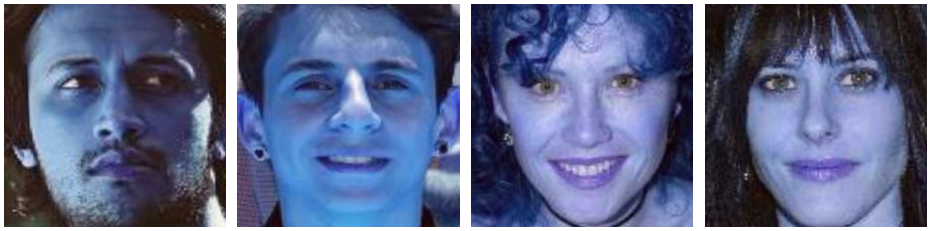

Fig. 3. Examples of base images

Fig. 4. Examples of adversarial attack images with glasses

Our results indicate that the adversarial attacks in the context of image classification can be easily used to hide sensitive information (e.g., gender information) that can be inferred by the image classification models.

4.2 Methodology and Experimentation for German Credit Dataset

Dataset creation
For evaluating our framework on discrete data, we chose Statlog (German Credit Dataset) [11]. The German Credit dataset has clearly defined attributes with respect to the ground truth. The dataset, however, is highly imbalanced with 70% of the attributes being good credit and 30% being bad credit and this imbalance needs to be handled with over/under sampling techniques. For the preprocess step, we encode the categorical features with one-hot encoding and normalize the numerical attributes. After the preprocessed data is fed to the pipeline, we handle the data imbalance by first over-sampling using SMOTE [7] and then under-sampling using the Edited Nearest Neighbours [32] technique. We train our models on this pipeline using 10-fold cross validation. Table 2 lists the best case validation accuracy of the select ML models on the German Credit dataset:

Table 2. Complete Training Data balanced with SMOTEENN

Classifier	Validation Accuracy	F1 score
RandomForest	76%	84%
AdaBoost	73%	82%
XGBoost	75%	84%
SVM	75%	83%
RidgeClassifier	72%	80%

Attack Mechanism
We choose 125 samples from the original test data that have been correctly classified as "Bad Credit". Our objective is to find the minimum cost multi-attribute change that will flip the classification of our examples to "good credit". We start by changing only one attribute at a time. After that we keep adding other attributes to be changed simultaneously. For example, in our first pass, we modify only attribute i_1 and record which samples change their classification. In the second pass of our algorithm, we change attributes i_1 and i_2 together and record the flipped samples. In the nth pass, we will be changing n attributes $i_1 \dots i_n$. We are only allowed to change an attribute from one of it's legitimate domain values to another. At each pass, we also record the transformation tuple set that caused target classification. Suppose we are changing two attributes i_1 (with subclasses j and k) and i_2 (with m and n). A possible transformation instance for n = 2 attributes may look like: $((A_j^{i_1}, A_k^{i_1}),(A_m^{i_2}, A_n^{i_2}))$. We discuss in detail about our cost functions for feasible transformations in the next section.

A transformation tuple based attack is similar to the adversarial example creation for images, in the sense that we cause imperceptible changes to the pixel values of our image to change the classification of our model. The difference between the two is that in the discrete scenario we change our data one attribute at a time: initially changing one attribute and recording if the classification changes, then, changing two attributes simultaneously and recording the classification change and so forth. Our algorithm is model agnostic and does not depend on the ML model's internal loss function formulation to work.

In this setting, we define the feasible instance space that includes only the following six modifiable attributes: Duration, Credit_amount, Purpose, Savings, Other_installment_plans, and Telephone. Since our algorithm involves multi-step multi-attribute change, the order of attribute change has impact on both speed and efficacy. We prefer the more sensitive attributes (attributes that easily cause change in the classification) to be changed early in our algorithm to ensure that we have the minimum attribute change for our examples. A simple way to decide the sensitivity of the attributes is to change each of the attributes individually and see which attributes have the highest attack success rate. Table 3 shows the attack success rate for our six attributes ordered from the highest success rate to the lowest success rate.

Table 3. Success rate of flipping classification result when one attribute is changed (out of 125)

Attribute	Attack success rate
purpose	20%
duration	10.4%
savings	8.8%
credit_amount	6.4%
telephone	4.8%
other_installment_plans	4%

Given the ordering of the attributes as shown in the table above, we run the multi-pass attack. We store those transformation tuples that cause the model to flip classification from "Bad" credit to "Good" credit. Table 4 gives us the attack success rate when we change more than one attribute. As we can see in the results, as the number of attributes changed increases, the attack success rate also increases. When we change six attributes, our attack success rate is 90%. However, this doesn't capture the cheapest possible attribute change for any given example which will be described in the next section.

Cost Formulation

Given a list of transformations that can be performed on an example, we also have the constraint that our attribute-changes should be drawn from a pool of feasible and ethical attribute modifications. To get a list of feasible transformations for each of the attributes, please refer to the appendix. To address the feasibility

Table 4. Success rate of number of examples flipped (out of 125)

Num attr changed	Attack success rate
2	52.8%
3	60.8%
4	69.64%
5	80.8%
6	90.4%

of individual attribute changes, we formulate a cost matrix C and assign a cost penalty to every attribute change. This cost penalty will be extremely large (close to infinity ∞) to discourage certain attribute change and certain sub-attribute changes. For other changes, the cost matrix formulation assigns small non-negative float value (between 0 and 1) as the cost. For example, for attribute i (with three subclasses j, k and z), the feasible attribute changes are: j to k, k to z, then $C^i_{j,z}$, $C^i_{z,k}$, $C^i_{k,j}$ and $C^i_{z,j}$ are all ∞ since they are infeasible changes. We also assign a weight w_i to each attribute i to weigh the influence of that particular attribute in our cost formulation. Assume, each example e can have a set of transformation tuples $M = (m_1, m_2, ...)$ such that $f(M(x)) = t$, where t is good credit. Given our cost formulation mechanism, (C and w), the cost required to get a classification flip from bad credit to good credit is $\arg\min_{\bigcup_{k=1}^{K}(\{i_k,j_k\})} \sum_{k=1}^{K} w_k \cdot C^k_{i_k,j_k}$, where K is the total number of attributes. We have three different cost formulation mechanisms for C and w that will be discussed below.

To understand the impact of the cost function, we experimented with three types of cost functions for the cost matrix formulation of attribute change. In all the three formulations, the infeasible attribute changes are assigned ∞ cost. The first cost function f_1 treats every feasible attribute change as equal. For example, if in one of our transformation tuple M, we are changing attribute i from subclass j to subclass k and attribute l from subclass m to n, then our cost function will ensure $C^i_{j,k} = C^l_{m,n}$ and $w_i = w_l$. The second cost function f_2 treats different attribute change differently, however each individual attribute will have a fixed cost for changes within the sub-classes. Going back to our example of attribute i with three subclasses (i, j and z), if the feasible modifications for i are j to k, and k to z, then $C^i_{j,k} = C^i_{k,z}$. However, for different attributes i and l, $C^i_{j,k} \neq C^l_{m,n}$ and $w_i \neq w_l$. The third cost function f_3 treats every attribute and sub-attribute change independently. The motivation behind this cost function formulation is that it might be easy to move between specific changes in sub-attribute classes for the same attribute class compared to others. So in the third case, $C^i_{j,k} \neq C^i_{j,z}$, $C^i_{j,k} \neq C^l_{m,n}$ and $w_i \neq w_l$. Figure 5 shows the relationship between minimum attributes required to be changed and the percentage of examples that can be flipped. We have plotted this comparison for our three different cost formulations. As we can see the distribution of the percentage has flattened with the introduction of variable weighting component into cost function formulation. Figure 6 gives the percentage of examples flipped as a function of the maximum cost possible (i.e., the maximum allowed cost of

changing all the feasible attributes without considering transformations with infinite costs). The fixed cost formulation has a very bumpy and uneven plot. As we introduce attribute weighting and non-uniform cost formulation for attribute changes, the graph becomes more smooth. As expected, as the "transformation cost" increases, more of the instances can flipped.

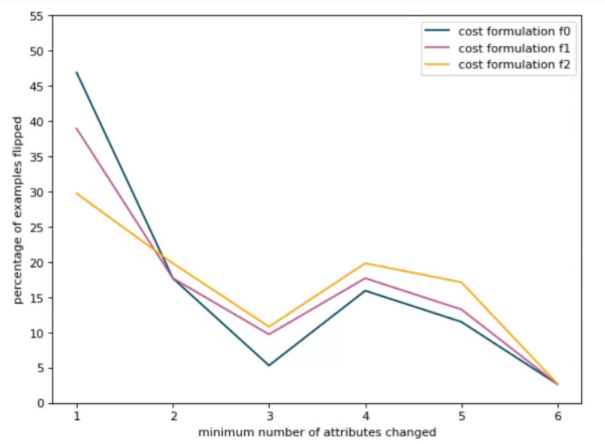

Fig. 5. Percentage of total attributes flipped vs min. attributes changed)

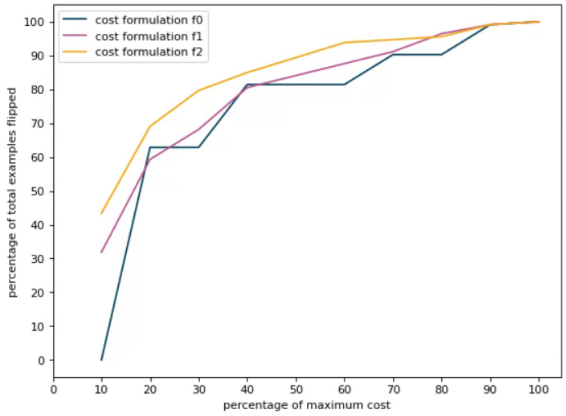

Fig. 6. Percentage of total attributes flipped vs percentage of maximum cost

5 Conclusion

In this paper, we present an approach to protecting individuals' privacy and fair opportunity by encouraging the end user to "game" a machine learning system in a legitimate manner. The idea is adapted from the adversarial learning problem that studies the vulnerability of machine learning systems to adversarial attacks—modifying data to foil the learning system. By incentivizing positive changes to the user's data profile, we can "convince" the learning system to make a different but fairer decision. If used properly, we show that this hostility against machine learning systems can become a powerful tool at the end user's disposal to protect and improve privacy and fairness. Our empirical results indicate that this idea can be successfully used, in a constrained way, to protect individuals against potentially harmful biases embedded in ML systems.

References

1. Akhtar, N., Mian, A.: Threat of adversarial attacks on deep learning in computer vision: A survey. IEEE Access **6**, 14410–14430 (2018)
2. Alufaisan, Y., Marusich, L.R., Bakdash, J.Z., Zhou, Y., Kantarcioglu, M.: Does explainable artificial intelligence improve human decision-making? (2020)
3. Ballet, V., Renard, X., Aigrain, J., Laugel, T., Frossard, P., Detyniecki, M.: Imperceptible Adversarial Attacks on Tabular Data. arXiv e-prints arXiv:1911.03274 (Nov 2019)
4. Bruckner, M., Scheffer, T.: Stackelberg games for adversarial prediction problems. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM (2011)
5. Carlini, N., Wagner, D.: Adversarial examples are not easily detected: Bypassing ten detection methods. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. p. 3Äì14. AISec Äô17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3128572.3140444, https://doi.org/10.1145/3128572.3140444
6. Carlini, N., Wagner, D.A.: Towards evaluating the robustness of neural networks. 2017 IEEE Symposium on Security and Privacy (SP) pp. 39–57 (2017)
7. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. J. Artif. Int. Res. **16**(1), 321–357 (Jun 2002)
8. Chouldechova, A., Roth, A.: A snapshot of the frontiers of fairness in machine learning. Commun. ACM **63**(5), 82–89 (Apr 2020). https://doi.org/10.1145/3376898, https://doi.org/10.1145/3376898
9. Commission, E.: 2018 reform of eu data protection rules
10. Dalvi, N., Domingos, P., Mausam, Sanghai, S., Verma, D.: Adversarial classification. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 99–108. KDD '04, ACM, New York, NY, USA (2004)
11. Dua, D., Graff, C.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
12. Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., Song, D.: Robust Physical-World Attacks on Deep Learning Visual Classification. In: Computer Vision and Pattern Recognition (CVPR)

13. Goodfellow, I., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: International Conference on Learning Representations (2015)
14. Hashemi, M., Fathi, A.: Permuteattack: Counterfactual explanation of machine learning credit scorecards (2020)
15. Ji, Z., Lipton, Z.C., Elkan, C.: Differential privacy and machine learning: a survey and review. CoRR **abs/1412.7584** (2014), http://arxiv.org/abs/1412.7584
16. Kanerva, A., Helgesson, F.: On the Use of Model-Agnostic Interpretation Methods as Defense Against Adversarial Input Attacks on Tabular Data. Master's thesis, , Department of Computer Science (2020)
17. Kantarcioglu, M., Xi, B., Clifton, C.: Classifier evaluation and attribute selection against active adversaries. Data Min. Knowl. Discov. **22**, 291–335 (January 2011)
18. Kulynych, B., Overdorf, R., Troncoso, C., Gürses, S.: Pots: Protective optimization technologies. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency. p. 177–188. FAT* '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3351095.3372853, https://doi.org/10.1145/3351095.3372853
19. Levin, S., Toerper, M., Hamrock, E., Hinson, J.S., Barnes, S., Gardner, H., Dugas, A., Linton, B., Kirsch, T., Kelen, G.: Machine-learning-based electronic triage more accurately differentiates patients with respect to clinical outcomes compared with the emergency severity index. Annals of Emergency Medicine **71**(5), 565 – 574.e2. https://doi.org/https://doi.org/10.1016/j.annemergmed.2017.08.005, http://www.sciencedirect.com/science/article/pii/S0196064417314427
20. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV) (December 2015)
21. Lowd, D., Meek, C.: Adversarial learning. In: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining. pp. 641–647. KDD '05 (2005)
22. Luo, J., Bai, T., Zhao, J., Li, B.: Generating adversarial yet inconspicuous patches with a single image (2020)
23. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. pp. 506–519. ASIA CCS '17, ACM, New York, NY, USA (2017)
24. Renard, X., Laugel, T., Lesot, M.J., Marsala, C., Detyniecki, M.: Detecting Potential Local Adversarial Examples for Human-Interpretable Defense. In: Workshop on Recent Advances in Adversarial Learning (Nemesis) of the European Conference on Machine Learning and Principles of Practice of Knowledge Discovery in Databases (ECML-PKDD). Dublin, Ireland (Sep 2018), https://hal.sorbonne-universite.fr/hal-01905948, presented at 2018 ECML/PKDD Workshop on Recent Advances in Adversarial Machine Learning (Nemesis 2018), Dublin, Ireland
25. Rudin, C., Wang, C., Coker, B.: The age of secrecy and unfairness in recidivism prediction. Harvard Data Science Review (1) (3 2020). https://doi.org/10.1162/99608f92.6ed64b30
26. Sharif, M., Bhagavatula, S., Bauer, L., Reiter, M.K.: Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In: Proceedings of the 23rd ACM SIGSAC Conference on Computer and Communications Security (2016)
27. Sharif, M., Bhagavatula, S., Bauer, L., Reiter, M.K.: A general framework for adversarial examples with objectives. ACM Transactions on Privacy and Security (2019)

28. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. In: International Conference on Learning Representations (2014), http://arxiv.org/abs/1312.6199
29. Vedaldi, A., Lenc, K.: Matconvnet – convolutional neural networks for matlab. In: Proceeding of the ACM Int. Conf. on Multimedia (2015)
30. Wang, Y., Kosinski, M.: Deep neural networks are more accurate than humans at detecting sexual orientation from facial images. (Oct 2018), osf.io/zn79k
31. Wenger, E., Passananti, J., Yao, Y., Zheng, H., Zhao, B.Y.: Backdoor attacks on facial recognition in the physical world. CoRR **abs/2006.14580** (2020), https://arxiv.org/abs/2006.14580
32. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man, and Cybernetics pp. 408–421 (1972)
33. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada. pp. 3320–3328 (2014), http://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks
34. Zhou, Y., Kantarcioglu, M.: Modeling adversarial learning as nested Stackelberg games. In: Bailey, J., Khan, L., Washio, T., Dobbie, G., Huang, J.Z., Wang, R. (eds.) Advances in Knowledge Discovery and Data Mining - 20th Pacific-Asia Conference, PAKDD 2016, Auckland, New Zealand, April 19-22, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9652, pp. 350–362. Springer (2016)
35. Zhou, Y., Kantarcioglu, M., Thuraisingham, B., Xi, B.: Adversarial support vector machine learning. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, New York, NY, USA (2012)

Appendix A

We consider the following attributes to change in our German Credit data:

1. Purpose: For getting the loan ex. car(new), car(old), repairs, education, etc.
2. Duration: Increase/decrease the duration (in months) to see the change in granting loan.
3. Credit amount: Increase and decrease the credit amount granted as a matter of percentage of original amount. ex: 1.05x, 1.10x, 0.90x, 0.85x where x is the current amount.
4. Savings account/bonds: Change the number of savings and bonds from None (A65) to '...100 DM' (A61).
5. Other installment plans: Change from None (A143) to Bank/Store (A141/A142).
6. Telephone: Change the ownership of telephone from None (A191) to registered in user's name (A192).