nature machine intelligence

ARTICLES

https://doi.org/10.1038/s42256-020-0216-z

Remote explainability faces the bouncer problem

Erwan Le Merrer[1] ✉ and Gilles Trédan[2] ✉

The concept of explainability is envisioned to satisfy society's demands for transparency about machine learning decisions. The concept is simple: like humans, algorithms should explain the rationale behind their decisions so that their fairness can be assessed. Although this approach is promising in a local context (for example, the model creator explains it during debugging at the time of training), we argue that this reasoning cannot simply be transposed to a remote context, where a model trained by a service provider is only accessible to a user through a network and its application programming interface. This is problematic, as it constitutes precisely the target use case requiring transparency from a societal perspective. Through an analogy with a club bouncer (who may provide untruthful explanations upon customer rejection), we show that providing explanations cannot prevent a remote service from lying about the true reasons leading to its decisions. More precisely, we observe the impossibility of remote explainability for single explanations by constructing an attack on explanations that hides discriminatory features from the querying user. We provide an example implementation of this attack. We then show that the probability that an observer spots the attack, using several explanations for attempting to find incoherences, is low in practical settings. This undermines the very concept of remote explainability in general.

Modern decision making driven by blackbox systems now impacts much of our lives[1,2]. These systems build on user data and range from recommenders[3] (for example, for personalized ranking of information) to predictive algorithms (for example, for credit default likelihood)[1]. This widespread deployment, along with the opaque decision processes of these systems, raises concerns about transparency for the general public and policy makers[4]. This has translated, in some jurisdictions (for example, in the United States and Europe), into a so-called 'right to explanation[4,5], which states that the output decisions of an algorithm must be motivated.

An already large body of work has explored the explainability of implicit machine learning models (such as neural network models)[6–8]. Indeed, these models show state-of-the-art performance when it comes to task accuracy, but they are not designed to provide explanations—or at least intelligible decision processes—when one wants to obtain more than the output decision of the model. In the context of recommendation, the expression 'post hoc explanation' has been coined[9]. In general, current techniques regarding the explainability of implicit models take trained in-house models and aim to shed light on some input features that cause salient decisions in their output space. LIME[10] (local interpretable model–agnostic explanations), for example, builds a surrogate model of a given blackbox system that approximates predictions around a region of interest. The surrogate is created from a newly crafted dataset, obtained from permutation of the original dataset values around the zone of interest and observation of the decisions made for this dataset). This surrogate is an explainable model by construction (such as a decision tree), so it can explain the decision that follows from a specific input. The number of queries to the blackbox model is assumed to be unbounded by LIME and other systems[11,12], permitting virtually exhaustive queries. This reduces their applicability to the inspection of in-house models by their designers.

As suggested by Andreou and others[13], some institutions could apply the same reasoning to explain some decisions to their users. Indeed, this would support the public's desire for a more transparent and trusted web. Facebook, for example, attempted to offer a form of transparency for the advertisement (ad) mechanism targeting its users by introducing a 'Why am I seeing this ad?' button on received ads. For users, the decision-making system (here, responsible for selecting relevant ads) is remote, and can be queried only by using inputs (their profile data). Yet, from a security standpoint (a security model where the remote server (executing the service) is untrusted by the users is considered in ref. [14]), Andreou and colleagues[13] empirically observed in the case of Facebook that these explanations are 'incomplete and can be misleading', conjecturing that malicious service providers can use this incompleteness to hide the true reasons behind their decisions.

In this Article, we question the possibility of such an explanation set-up, where a corporate and private model is issuing decisions to users. We go one step further by demonstrating that remote explainability simply cannot be a reliable guarantee of a lack of discrimination in the decision-making process. In a remote blackbox set-up such as that of Facebook, we show that a simple attack, which we coin the 'public relations' (PR) attack, undermines remote explainability.

For demonstration purposes we introduce the 'bouncer problem' as an illustration of the difficulty for users to spot malicious explanations. The analogy works as follows: let us picture a bouncer at the door of a club, deciding who may enter. When he issues a negative decision—refusing entrance to a given person—he also provides an explanation for this rejection. However, his explanation might be malicious, in the sense that his explanation does not present the true reasons for the rejection. Consider, for example, a bouncer discriminating people based on the colour of their skin. Of course he will not tell people he is refusing them entrance based on that characteristic, because this is a legal offence. He will, instead, invent a biased explanation that the rejected person is likely to accept.

The classic way to assess discrimination by a bouncer is for associations to run tests (following the principle of statistical causality[15], for example): several persons who attempt to enter vary only in their attitude or appearance in terms of the possibly discriminating feature (such as the colour of their skin). Conflicting decisions by the bouncer are then an indication of a possible discrimination, supporting building a case for prosecution.

[1]Inria Rennes – Bretagne Atlantique Research Centre, Rennes, France. [2]French National Centre for Scientific Research, Paris, France.
✉e-mail: erwan.le-merrer@inria.fr; gtredan@laas.fr

We make a parallel with bouncer decisions in this Article by demonstrating that a user cannot trust a single (one-shot) explanation provided by a remote model. Moreover, we show that creating such malicious explanations necessarily creates inconsistent answers for some inputs, and the only way to spot those inconsistencies is to issue multiple requests to the service. Unfortunately, we also demonstrate the problem to be hard, in the sense that spotting an inconsistency in such a way is intrinsically not more efficient than for a model creator to exhaustively search on her local model to identify a problem, which is often considered an intractable process.

In the next section we build a general set-up for remote explainability, which has the purpose of representing actions by a service provider and by users facing model decisions and explanations. The fundamental blocks for observation of the impossibility of a reliable remote explainability or its hardness for multiple queries are then presented. We next present the bouncer problem, which users have to solve to detect malicious explanations by the remote service provider. We then illustrate the PR attack, which the malicious provider may execute to remove discriminative explanations to users, on decision trees. The bouncer problem is addressed practically by modelling a user trying to find inconsistencies from provider decisions based on the German Credit dataset and a neural network classifier. We also discuss open problems, before reviewing related works and conclusions. Because we show that remote explainability in its current form is undermined, this work aims to motivate researchers to explore the direction of provable explainability by designing new protocols (such as with cryptographic means, for example, in proof of ownership for remote storage) or to build collaborative observation systems to spot inconsistencies and malicious explanation systems.

Explainability of remote decisions

In this Article, we study classifier models, which issue decisions given user data. We first introduce the set-up in which we operate, which is intended to be as general as possible, so that the results drawn from it can apply widely.

General set-up. We consider a classifier $C : \mathcal{X} \mapsto \mathcal{Y}$ that assigns inputs x of the feature space $\mathcal{X}$ to a class $C(x) = y \in \mathcal{Y}$. Without loss of generality and to simplify the presentation, we will assume the case of a binary classifier: $\mathcal{Y} = \{0, 1\}$; the decision is thus the output label returned by the classifier.

Discriminative features and classifiers. To produce a decision, classifiers rely on features as an input. These are, for instance, the variables associated to a user profile on a given service platform (for example, basic demographics, political affiliation, purchase behaviour and residential profile[13]). In our model, we consider that the feature space contains two types of feature: discriminatory and legitimate. The use of discriminatory features allows for exhibiting the possibility of a malicious service provider issuing decisions and biased explanations. This problematic is also referred to as 'rationalization' in ref. [16].

Concretely, and for the sake of our demonstration, we consider discriminatory features to be an arbitrary subset of the input features, such that we can define these as 'any feature set the malicious service provider does not want to explain'. Two main reasons come to mind:

- Legal: the jurisdiction's law forbids decisions based on a list of criteria (for example, in the UK, see https://www.gov.uk/discrimination-your-rights). A service provider risks prosecution on admitting the use of these. For instance, features such as age, sex, employment or the status of foreigner are considered as discriminatory in ref. [17]; this examines the German Credit dataset, which links bank customer features to the accordance or not of a credit.
- Strategic: the service provider wants to hide the use of some features on which its decisions are based. This could be to hide some business secret from competitors (for instance, because of the accuracy–fairness trade-off[18]), to avoid 'reward hacking' from users biasing this feature or simply to avoid bad press.

- Conversely, any feature that is not discriminatory is coined 'legit'.

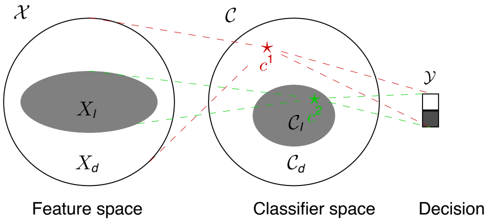

Fig. 1 | Illustration of our model. We consider binary classifiers, which map the input domain $\mathcal{X}$ to labels $\mathcal{Y} = \{0, 1\}$. Some dimensions of the input space are discriminative, X_d, which induces a partition on the classifier space. Legitimate classifiers $\mathcal{C}_l$ do not rely on discriminative features to issue a label (in green), while others (that is, $\mathcal{C}_d$) can rely on any feature (in red).

Formally, we partition the classifier input space $\mathcal{X}$ along these two types of feature: legitimate features X_l that the model can legitimately exploit to issue a decision and discriminatory features X_d (Fig. 1). In other words $\mathcal{X} = (X_l, X_d)$, and any input $x \in \mathcal{X}$ can be decomposed as a pair of legitimate and discriminatory features $x = (x_l, x_d)$ (the introduction of such a split in the features is required to build our analysis). We assume the input contains at least one legitimate feature: $X_l \neq \emptyset$.

We also partition the classifier space accordingly: let $\mathcal{C}_l \subset \mathcal{C}$ be the space of legitimate classifiers (among all classifiers $\mathcal{C}$) that do not rely on any feature of X_d to issue a decision. More precisely, we consider that a classifier is legitimate if and only if arbitrarily changing any discriminative input feature never changes its decision:

$$C \in \mathcal{C}_l \iff \forall x_l \in X_l, \forall x_d, x'_d \in \mathcal{X}_d^2, C((x_l, x_d)) = C((x_l, x'_d))$$

Observe, therefore, that any legitimate classifier C_l can simply be defined over input subspace $X_l \subset \mathcal{X}$. As a slight notation abuse to stress that the value of discriminative features does not matter in this legitimate context, we write $C((x_l, \emptyset))$, or $C(x \in X_l)$, as the decision produced, regardless of any discriminative feature. It follows that the space of discriminative classifiers complements the space of legitimate classifiers: $\mathcal{C}_d = \mathcal{C} \setminus \mathcal{C}_l$.

We can now reframe the main research question we address: Given a set of discriminative features X_d, and a classifier C, can we decide if $C \in \mathcal{C}_d$ in the remote blackbox interaction model?

The remote blackbox interaction model. We question the 'remote blackbox interaction' model (see, for example, ref. [19]), where the classifier is exposed to users through a remote application programming interface (API). Users can only query the classifier model with an input and obtain a label as an answer (for example, 0 or 1). In this remote set-up, users then cannot collect any specific information about the internals of the classifier model, such as its architecture, its weights or its training data. This corresponds to a security threat model where two parties are interacting with each other (the user and the remote service) and where the remote model is implemented on a server belonging to the service operator, which is untrusted by the user.

Requirements for remote explainability. Explainability is often presented as a solution to increase the acceptance of artificial intelligence (AI)[6] and to potentially prevent discriminative AI behaviour. Let us expose the logic behind this connection.

Explanations using conditional reasoning. First, we need to define what is an explanation, to go beyond Miller's definition as an 'answer to a why-question'[20]. Because the topic of explainability is becoming a hot research field with (to the best of our knowledge) no consensus on a more technical definition of an explanation, we will propose, for the sake of our demonstration, that an explanation is causally coherent with respect to the modus ponens rule from deductive reasoning ('if A is implying B, and A being true, B is true as well')[21]. For instance, if explanation a explains decision b, it means that in context a, the decision produced will necessarily be b. In this light, we first directly observe the beneficial effect of such explanations on our parallel to club bouncing. While refusing someone, the bouncer may provide reasons for that rejection. The person can then change their behaviour so as to be accepted on the next attempt.

Second, this modus ponens explanation form is also sufficient to prove non-discrimination. For instance, if a does not involve discriminating arguments (which can be checked by the user as a is a sentence), and $a \Rightarrow b$, then decision b is not discriminative. On the contrary, if a does involve discriminating arguments (for simplicity, we leave aside the fact that some features might proxy some others[22]—in particular discriminative ones; indeed, if even when listing all features, one cannot assess discrimination, then the problem is even harder due to correlations and those proxies), then decision b is taken on a discriminative basis and is therefore a discriminative decision. In other words, this property of an explanation is enough to reveal discrimination.

To sum up, any explanation framework that behaves 'logically' (that is, fits the modus ponens[21])—which is in our view a rather mild assumption—is enough to establish the discriminative basis of a decision. We believe this is the rationale of the statement 'transparency can improve users trust in AI systems. In fact, this logical behaviour is not only sufficient to establish discrimination, it is also necessary: assume a framework providing explanation a for decision b such that we do not have $a \Rightarrow b$. Because a and b are not connected anymore, a does not bring any information about b.

Although this logical behaviour is desirable for users, unfortunately, in a remote context they cannot check whether $a \Rightarrow b$ is in general true because they are only provided with a particular explanation a leading to a particular decision b. They cannot check that a being true leads in all contexts to b being true.

Requirements on the user side for checking explanations. In a nutshell, a user in a remote interaction can verify that in her context a is true and b is true, which is compatible with the $a \Rightarrow b$ relation of an explanation fitting the modus ponens. Let us formalize what a user can check regarding the explanation she collects. A user who queries a classifier C with an input x gets two elements: the decision (inferred class) $y = C(x)$ and an explanation a such that a explains y. To produce such explanations, we assume the existence of an explanation framework exp_C producing explanations for classifier C (this could, for instance, be the LIME framework[10]). Formally, upon request x, a user collects y and $a = exp_C(y,x)$ explaining decision y in context x by classifier C.

We assume that such a user can check that a is apropos (that is, appropriate): a corresponds to input x. We write $a \in A(x)$. This allows us to formally write a non-discriminatory explanation as $a \in A(x_l)$. This forbids lying by explaining an input that is different from x.

We also assume that the user can check the explanation is consequent: the user can check that a is compatible with y. This forbids crafting explanations that are incoherent w.r.t. the decision (like a bouncer explaining why you can enter while leaving the door locked).

Having defined the considered model for constructing and exposing our results, we stress that this model aims at constraining the provider as much as possible. In particular, explanations must be as complete as possible, must always be provided and must always be coherent with the decision. The intuition is that if we prove the possibility of malicious explanations even in this constrained case, then in all less constrained cases (such as for incomplete explanations as observed in ref. [13] or example-based explanations), the trickery will only be easier.

To sum up for the explanation model: explanations fitting the modus ponens allow users to detect discrimination. Unfortunately, in a remote context, users cannot check whether explanations do fit the modus ponens. However, they can check the veracity of the explanation and the decision in their particular experience. This is the space we exploit for our attack, by generating malicious explanations that appear correct to the user (but that do not fit the modus ponens).

The PR attack or the limits of remote explainability. We articulate our demonstration of the limits of explainability in a remote set-up by showing that a malicious service provider can hide the use of discriminating features for issuing its decisions, while conforming to the mild explainability framework we described.

Such a malicious provider thus wants to (1) produce decisions based on discriminative features and (2) produce non-discriminatory explanations to avoid prosecution. A first approach could be to manipulate the explanation directly. However, it might be difficult to do so while keeping the explanation convincing and true in an automated way. In this Article, we follow another strategy that instead consists in creating a legitimate classifier that will then be explained.

A generic attack against remote explainability. We coin this attack the public relations (PR) attack. The idea is rather simple: on reception of an input x, first compute the discriminative decision $C(x)$. Then train a surrogate model C' that is non-discriminative and such that $C'(x) = y$. Explain $C'(x)$, and return this explanation along with $C(x)$.

Figure 2 illustrates a decision based solely on legitimate features (Fig. 2a), a provider giving an explanation that includes discriminatory features (Fig. 2b) and the attack by a malicious provider (Fig. 2c). In all three scenarios, a user is querying a remote service with inputs x, and obtaining decisions y, each along with an explanation. In the case of Fig. 2b, the explanation exp_C reveals the use of discriminative features X_d; this provider is prone to complaints. To avoid these, the malicious provider (Fig. 2c) leverages the PR attack, by first computing $C(x)$ using its discriminative classifier C. Then, based on the legitimate features x_l of the input, and its final (discriminative) decision y, it derives a classifier C' for the explanation. Core to the attack is the ability to derive such a classifier C'.

Informally, coherence ensures that the explanation (derived from C') appears consequent to the user observing decision y, while legitimacy ensures that the explanation will appear to the user as originating from the modus ponens explanation of a non-discriminating classifier.

Effectiveness of the attack. Let us consider the perspective of a user who, upon request x, collects a y answer along with an explanation a. Observe that $a = exp_{C'}(y, x)$ is apropos because it directly involves x: $a \in A(x)$. Because we have $C'(x) = y$ it is also consequent. Finally, given that $C' \in \mathcal{C}_l$, then $a \in A(x_l)$: a is non-discriminatory. So, from the user perspective, she collects an apropos and consequent explanation that could originate from the logical explanation of a legitimate classifier.

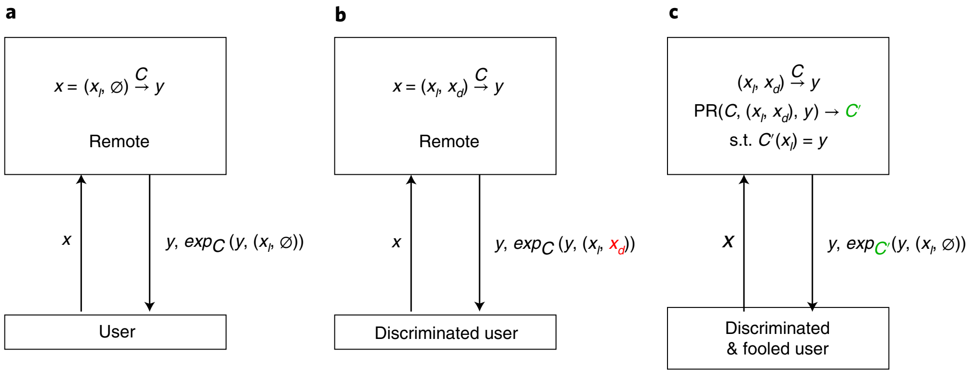
a
$x = (x_l, \varnothing) \xrightarrow{C} y$
Remote
x
y, $exp_C\,(y, (x_l, \varnothing))$
User
b
$x = (x_l, x_d) \xrightarrow{C} y$
Remote
x
y, $exp_C\,(y, (x_l, x_d))$
Discriminated user
c
$(x_l, x_d) \xrightarrow{C} y$
$PR(C, (x_l, x_d), y) \to C'$
s.t. $C'(x_l) = y$
x
y, $exp_{C'}(y, (x_l, \varnothing))$
Discriminated & fooled user

Fig. 2 | The three scenarios involving remote explainability. **a**, A provider using a model that does not leverage discriminatory features. **b**, A discriminative model divulges its use of a discriminating feature. **c**, The PR attack principle, undermining remote explainability: the blackbox builds a surrogate model C' for each new request x, which decides y based on x_l features only. It explains y using C'.

Existence of the attack. We note that crafting a classifier C' satisfying the first property is trivial because it only involves a single data point x. An example solution is the Dirac delta function of the form

$$C'(x') = C'((x'_l, x'_d)) = \begin{cases} \delta_{x'_l, x_l} & \text{if } y = 1 \\ 1 - \delta_{x'_l, x_l} & \text{if } y = 0 \end{cases}$$

where δ is the Dirac delta function. Informally, this solution corresponds to defining the classifier that would only answer bounce to this specific input x, and answer enter to any other input.

Although a corresponding intuitive explanation could be 'because it is specifically you', explaining this very specific function might not fit any explainability framework. To alleviate this concern, we provide an example implementation of a PR attack that produces legitimate decision trees from discriminating ones in the section 'Illustration using decision trees'.

Dirac here only constitutes an example proving the existence of PR attack functions. It is important to realize that many such C' PR attack functions exists (any function $X_l \mapsto \mathcal{Y}$ that satisfies the easy $C'(x) = y$ condition).

In other words, PR attack functions are easy to find: if one could sample $\mathcal{C}_l$ uniformly at random, because $C'(x) = y$ is equally likely as $C'(x) = \bar{y}$, each sample would yield a PR attack function with probability 1/2.

We have presented the framework and an attack necessary to question the possibility of remote explainability. We next discuss the possibility for a user to spot that an explanation is malicious and obtained by a PR attack. We stress that if a user cannot, then the very concept of remote explainability is at stake.

Definition 1. PR attack. Given an arbitrary classifier $C \in \mathcal{C}_d$, a PR attack is a function that finds for an arbitrary input x a classifier C':

$$\mathrm{PR}\,(\mathrm{C}, \mathrm{x}, \mathrm{C}(\mathrm{x})) \to \mathrm{C}' \tag{1}$$

such that C' satisfies two properties:

- coherence: $C'(x_l) = y$
- legitimacy: $C' \in \mathcal{C}_l$

Proposition 1. Let $\mathcal{C}_l : X_l \mapsto \mathcal{Y} = \{0, 1\}$ be the set of all possible legit classifiers, and its cardinality be $|\mathcal{C}_l|$. Let $\mathcal{PR} \subset \mathcal{C}_l$ be the set of possible PR attack functions. We have $|\mathcal{PR}| = |\mathcal{C}_l|/2$: half of all possible legit classifiers are PR attack functions.

Proof. Pick $x_l \in X_l$ and $y = C(x)$ the decision with which our PR attack function must be coherent. Because $\mathcal{C}_l$ is a set of functions defined over X_l, any particular function C in $\mathcal{C}_l$ is defined at x_l. Let us partition the function space $\mathcal{C}_l$ according to the value these functions take at x_l: let $\mathcal{A} : \{C \in \mathcal{C}_l \text{ s.t. } C(x_l) = y\}$ and $\mathcal{B} : \{C \in \mathcal{C}_l \text{ s.t. } C(x_l) = \bar{y}\}$. We have $\mathcal{C}_l = \mathcal{A} \cup \mathcal{B}$. Let $not : \mathcal{A} \mapsto \mathcal{B}$ be a 'negation function' that associates for each function $C \in \mathcal{A}$ its negation $not(C) \in \mathcal{B}$ s.t. $not(C)(x) = 1 - C(x)$. Observe that $not \circ not = Id$: not defines a bijection between $\mathcal{A}$ and $\mathcal{B}$ (any function in A has exactly one unique corresponding function in $\mathcal{B}$ and vice versa). Because not is a bijection, we deduce $|\mathcal{A}| = |\mathcal{B}| = |\mathcal{C}_l|/2$. Given that $\mathcal{A}$ contains all possible legitimate functions ($\mathcal{A} \subset \mathcal{C}_l$) that are coherent with $C(x_l) = y$, $\mathcal{A} = \mathcal{PR}$. Thus $|\mathcal{PR}| = |\mathcal{C}_l|/2$. □

The bouncer problem

We presented in the previous section a general set-up for remote explainability. We now formalize our research question regarding the possibility of a user spotting an attack in that set-up.

Definition 2. The bouncer problem (BP). Using ϵ requests that each returns a decision $y_i = C(x_i)$ and an explanation $exp_C(y_i, x)$, decide if $C \in \mathcal{C}_d$. We denote that action by $BP(\epsilon)$.

An observation for one-shot explanations. As a first step analysis in this set-up, we show that an attack cannot be spotted in the case of one-shot explanations. We already know that using a single input point is insufficient.

Observation 1. $BP(1)$ has no solution.

Proof. The Dirac construction above always exists. □

Indeed, constructions like the introduced Dirac function form a PR attack that produces explainable decisions. Given a single explanation on model C' (that is, $\epsilon = 1$), a user cannot distinguish between the use of a model (C in Fig. 2, case (a)) or one of a model crafted by a PR attack (C' in Fig. 2, case (c)), because it is consequent. This means that such a user cannot spot the use of hidden discriminatory features due to the PR attack by the malicious provider.

We observed that a user cannot spot a PR attack with $BP(1)$. This is already problematic, as it gives a formal proof of why the Facebook ad explanation system cannot be trusted[13].

The hardness of multiple queries for explanation. To address the case $BP(\epsilon > 1)$, we observe that a PR attack generates a new model C' for each request; as a consequence, an approach to detect that attack is to detect the impossibility (using multiples queries) of a single model C' to produce coherent explanations for a set of observed decisions. Here, we study this approach.

Interestingly, classifiers and bouncers share this property that their outputs are all mutually exclusive (each input is mapped to exactly one class). Thus we have `enter`$\Rightarrow \overline{bounce}$ (with `enter` and `bounce` being the positive or negative decision to enter a place, for instance). In this case it is impossible to have $a \Rightarrow$`enter` and $a \Rightarrow$`bounce`. Note that this relation assumes a 'logical' explainer. On a non-logical explainer, because we cannot say $a \Rightarrow$`enter` given a and `enter`, we cannot detect such an attack. Note also that non-mutually exclusive outputs (for example, in the case of recommenders where recommending item a does not imply not recommending item b) are not bound by this rule.

A potential problem for the PR attack is a decision conflict, in which a could explain both b and $\bar{b}$, its opposite. For example, imagine a bouncer refusing you entrance to a club because, say, you have white shoes. Then, if the bouncer is coherent, he should refuse entrance to anyone wearing white shoes, and if you witness someone entering with white shoes, you could argue against the lack of coherence of the bouncer decisions. We build on these incoherences to spot PR attacks.

To examine the case $BP(\epsilon)$, where $\epsilon > 1$, we first define the notion of an 'incoherent pair' (IP).

Definition 3. Incoherent pair (IP). Let $x^1 = (x^1_{\rm l}, x^1_{\rm d}), x^2 = (x^2_{\rm l}, x^2_{\rm d}) \in \mathcal{X} = X_{\rm l} \times X_{\rm d}$ be two input points in the feature space (with × denoting the Cartesian product). x^1 and x^2 form an IP for classifier C if they both have the same legit feature values in $X_{\rm l}$ and yet end up being classified differently:

$x^1_{\rm l} = x^2_{\rm l} \wedge C(x_1) \neq C(x_2)$. For convenience we write $(x^1, x^2) \in IP_C$.

Finding such an IP is a powerful proof of PR attack on the model by the provider: only decisions resulting from a model crafted by a PR attack can exhibit IPs: $IP_C \neq \emptyset \Rightarrow C \in \mathcal{C}_{\rm d}$. Intuitively, this is a formalization of an intuitive reasoning: 'if you let others enter with white shoes then this was not the true reason for my rejection'.

We can show that there is always a pair of inputs allowing us to detect a discriminative classifier $C \in \mathcal{C}_{\rm d}$.

Proposition 2. A classifier C', resulting from a PR attack, always has at least one IP: $C' \in \mathcal{C}_{\rm d} \Rightarrow IP_C \neq \emptyset$.

Proof. We prove the contrapositive form $IP_C = \emptyset \Rightarrow C \notin \mathcal{C}_{\rm d}$. Informally, the strategy here is to prove that if no such pair exists, this means that decisions are not based on discriminative features in $X_{\rm d}$, and thus the provider had no interest in conducting a PR attack on the model; the considered classifier is not discriminating.

Assume that $IP_C = \emptyset$. Let $x_\emptyset \in X_{\rm d}$, and let $C^{\rm l} : X_{\rm l} \mapsto \mathcal{Y}$ be a legitimate classifier such that $C^{\rm l}(x_{\rm l}) = C((x_{\rm l}, x_\emptyset))$.

Since $IP_C = \emptyset$, this means that $\forall x^1, x^2 \in \mathcal{X}, x^1_{\rm l} = x^2_{\rm l} \Rightarrow C(x_1) = C(x_2)$. In particular $\forall x \in \mathcal{X}, C(x = (x_{\rm l}, x_{\rm d})) = C^{\rm l}(x_{\rm l}, x_\emptyset)$. Thus $C = C^{\rm l}$; by the definition of a PR attack being only applied to a model that uses discriminatory features, this leads to $C \in \mathcal{C} \setminus \mathcal{C}_{\rm d}$, that is, $C \notin \mathcal{C}_{\rm d}$. □

This directly applies to our problem.

Proposition 3. Detectability lower bound. $BP(|\mathcal{X}|)$ is solvable.

Proof. Straightforward: $C' \in \mathcal{C}_{\rm d} \Rightarrow IP_C \neq \emptyset$, and since $IP \subseteq \mathcal{X} \times \mathcal{X}$, testing the whole input space will necessarily exhibit such an IP. □

This last result is rather weakly positive. Even though any PR attack is eventually detectable, in practice it is impossible to exhaustively explore the input space of modern classifiers due to their dimension. This remark also further questions the opportunity of remote explainability.

Moreover, it is important to observe that while finding an IP proves the presence of a PR attack, it is not an efficient technique to prove the absence of a PR attack, which is probably the use case that most interests users. The section 'Finding IPs on a neural network model' details this approach on a concrete dataset.

This concludes the theoretical perspective of this Article. To sum up, an explainer that could allow spotting classifier discrimination should behave logically; this is what is expected by users. However, they can only check the properties of the provided explanation with regard to input x, which leaves room for malicious providers. One such provider can just create a legit explainer whose decision matches the discriminative one for input x. Fortunately, this technique can be detected. However, this detection is difficult in practice, as we will illustrate next.

Illustration and experiments

In this section, we instantiate concretely some of the points raised by our theoretical perspective. We first illustrate the ease of finding PR attack functions on binary decision trees by presenting an algorithm that implements one. We then focus on detection, and evaluate in practice the hardness of finding IPs on the German Credit dataset.

Illustration using decision trees. In this section, we embody the previous observations and approaches on the concrete case of decision tree classifiers. The choice of decision trees was motivated first because of its recognized importance (for example, with `C4.5` ranked number one of the top 10 data-mining algorithms[23]). Second, there is a wide consensus on their explainability, which is straightforward[8]: a path in the tree 'naturally' lists the attributes considered by the algorithm to establish a classification. Finally, the simplicity of crafting PR attacks on these makes them good candidates for an illustration and argues for the practical implementability of such an attack.

We denote $\mathcal{T}$ as the set of tree-based classifiers. We do not need any assumption on how the tree is built (for example, `C4.5`[24]). Regarding explainability, we here only need to assume that decision trees are explainable: $\forall C \in \mathcal{T}, exp_C$ exists.

Let $C \in \mathcal{T} \cap \mathcal{C}_{\rm d}$ be a discriminatory binary tree classifier. Each internal node $n \in V(C)$ tests incoming examples based on a feature *n.label*. Each internal node is connected to exactly two sons in the tree, named *n.right* and *n.left*. Depending on the (binary) result of this test, the example will continue on either of these paths. We denote the father of an internal node by *n.father* (the root node r is the only node, such that $r.father = \emptyset$).

Algorithm 1 presents a PR attack on binary decision trees. To ease its presentation, we assume that given an input x, *n.right* will by convention always be the branch taken after evaluating x on n. The algorithm starts by initializing the target decision tree C' as a copy of C. Then, it selectively removes all nodes involving discriminative features, and replaces them with the subtree the target example x would take.

Algorithm 1. PR attack on a discriminative decision binary tree C.

```
Input: $C, x = (x_l, x_d)$
$y = C(x)$;   // Find discriminative decision
Let $\{n_0, \ldots n_t\}$ be the breadth first ordering of the
   nodes of $C$;
Let $C' = C$;   // Initialise surrogate
for $node\ i = 0$ to $t$ do
   if $n_i$.label $\in X_d$ then
      $C'.n_i.father.right = n_i.right$;   // Reconnect
      $n_i$ father to right son
      $C' = C' \setminus \{n_i\}$;   // Remove discriminating
      node
      $C' = C' \setminus \{n_i.left$ subtree$\}$;   // Remove left
      subtree
   else
      $C'.n_i.left = \bar{y}$;   // Keep legit node, add
      dummy terminal node
   end
end
return $y, exp_{C'}(y, (x_l, \emptyset))$
```

To do so, Algorithm 1 removes each discriminative node n_i by connecting n_{i-1} and n_{i+1}. Although this approach would be problematic in the general case (we would lose the n_i*.left* subtree), in the context of x we know the explored branch is n_i*.right*, so

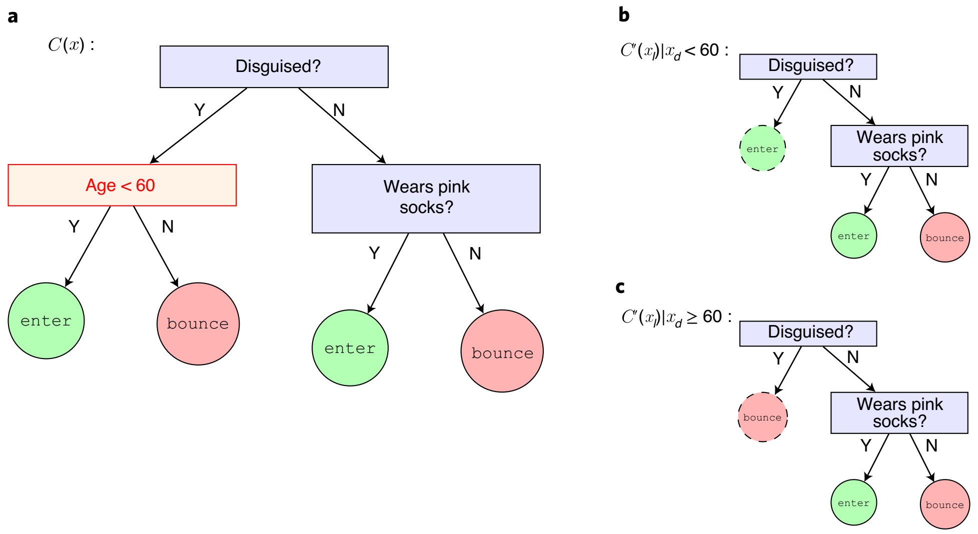

Fig. 3 | Illustration of a possible implementation (Algorithm 1) of the PR attack. a–c, Instead of having to explain the use of a discriminative feature (age in this case) in classifier C in **a**, two non-discriminative classifiers (C' in **b** and **c**) are derived. Depending on the age feature in the request, a C' is then selected to produce legit explanations. The two dashed circles represent an IP, which users might seek to detect a malicious explanation.

we simply reconnect this branch and replace the left subtree by a dummy output.

Proposition 4. Algorithm 1 implements a PR attack.

Proof. To prove the statement, we need to prove that

- C' is legitimate
- C' is coherent: $C'(x_l) = y$
- C' is explainable

First, observe that any nodes of C' containing discriminative features are removed (line 7). Thus, C' only takes decisions based on features in X_l: C' is legitimate.

Second, observe that, by construction, since $x = (x_l, x_d)$ and since any discriminative node n is replaced by this right ($n.right$) outcome, which is the outcome that corresponds to x_d. In other words, $\forall x'_l \in X_l, C'(x'_l) = C((x'_l, x_d))$: C' behaves like C where discriminative features are evaluated at x_d. This is true in particular for $x_l : C'(x_l) = C((x_l, x_d)) = C(x) = y$.

Finally, observe that C' is a valid decision tree. Therefore, according to our explainability framework, C' is explainable. □

Interestingly, the presented attack can be efficient as it only involves pruning part of the target tree. In the worst case, this one has $\Omega(2^d)$ elements, but, in practice, decision trees are rarely that big.

A simple example. An example is presented in Fig. 3 of a classifier C that uses three features—the fact that persons are disguised or not, the fact that they wear pink socks or not and their age—to decide whether or not they can enter a place. A malicious service provider using that discriminative classifier is queried using input x. This classifier is shown on the left and exploits a single discriminative (binary) dimension of x: 'Age < 60'.

Let us consider that exp is a canonical explanation framework that, given a decision tree C and an input x, translates into words the path taken by x through C down to the decision leaf. In such a framework, if x represents a disguised individual over 60, $C(x)$ is bounce and the explanation $exp_C(C(x), x)$ would be 'You cannot enter because we do not let in disguised people over 60'—corresponding to the path (Y,N) in C that effectively reveals the discrimination.

Instead of replying the explanation $exp_C(C(x),x)$, the malicious provider implements a PR attack as follows. It computes the discriminative decision $C(x)$, but generates an explanation using a classifier C' derived by removing the discriminative features of C (using Algorithm 1).

Using the same 'disguised over 60' input x in this PR attack context (1) yields the original decision $C(x)$ but (2) yields an explanation based on the Fig. 3c classifier $C'(x_l)|x_d \geq 60$. In this classifier, the path for x_l is only (Y), which is explained by exp as 'You cannot enter because we do not let in disguised people'.

As a second example, consider an x' that represents someone disguised but under 60. This input yields a different decision $C(x')$ (enter, path (Y,Y)). Using Algorithm 1 again, a new classifier C' is generated. This $C'(x'_l)|x'_d < 60$ is the Fig. 3b classifier, in which the path corresponding to x' is (Y), which is explained by exp as 'You can enter because we allow disguised people'. (In general, we cannot guarantee that the provided explanation makes sense to the user. Exploiting this information might provide additional room for detecting malicious explanations in some practical settings. However, because C' is derived from C, the provided explanation $exp_{C'}$ is probably as credible as exp_C to the user, unless C' is very different from C due to a high discrimination ratio.)

As shown, both x and x' yield non-discriminative explanations for a discriminative decision.

In this example, comparing both versions of C' easily yield solutions for $BP(2)$, for example (disguised, whitesocks, Age = 49) and (disguised, whitesocks, Age = 62).

PR attacks on purely discriminative classifiers. We now elaborate on an extreme case that is challenging for constructing a PR attack, namely a discriminative decision tree containing only discriminative nodes. For example, consider C: if (gender = male) then enter else bounce, being a discriminative classifier that rejects individuals based only on their gender. It can be represented by a decision tree containing a single discriminative node. As our input space contains only two possible values (male and female), Algorithm 1 generates only two legit classifiers $C'(x|x = male)$ leading to an enter decision, and $C'(x_d|x_d = female)$ leading to bounce. Both legit surrogates only contain a single decision node, which could correspond to explanations like 'We always let everyone in' and 'We never let anyone in'.

In this case, any mixed couple of inputs (female, male) constitutes an IP. Therefore, assuming female and male inputs are equally likely, the probability of a random pair of inputs to constitute an IP is 1/2. Although such a discriminative model is unlikely to be placed in production, this examples stresses the detection ease of PR attacks in that extreme case.

Finding IPs on a neural network model. We now take a closer look at the detectability of the attack, namely, how difficult is it to spot an IP?

For the experimental set-up we leveraged Keras over TensorFlow to learn a neural network-based model for the German Credit dataset[25]. Although we could have used any relevant type of classifier for our experiments, the general current focus is on neural networks for explainability. The bank dataset classifies client profiles (1,000 of them), described by a set of attributes, as good or bad credit risks. Multiple techniques have been employed to model the credit risks on that dataset, which range from 76.59% accuracy for a support vector machine to 78.90% for a hybrid between a genetic algorithm and a neural network[26].

The dataset is composed of 24 features (some categorical ones, such as sex, were set to numerical). This thus constitutes a low-dimensional dataset compared to current applications (observations in ref. [13] reported up to 893 features for the sole application of ad placement on user feeds on Facebook). Furthermore, modern classifiers are currently dealing with up to $512 \times 512 \times 3$ dimensions[27]; this shows a significant increase in data processing and thus the capability to expand the number of features taken into account for decision making.

Our neural network (code is available at https://github.com/erwanlemerrer/bouncer_problem) is inspired by the one proposed[28] in 2010, which reached 73.17% accuracy. It is a simple multilayer perceptron with a single hidden layer of 23 neurons (with sigmoid activations) and a single output neuron for binary classification of the input profile to 'risky' or not. In this experiment we use the Adam optimizer and a learning rate of 0.1 (leading to much faster convergence than in ref. [28]), with a validation split of 25%. We create 30 models, with an average accuracy of 76.97%@100 epochs on the validation set (with a standard deviation of 0.92%).

To generate input profiles, we consider two scenarios. In scenario A, a user sets a random value in a discriminative feature to try to find an IP. This yields rather artificial user profiles (which may be detected as such by the remote service provider). To obtain an aggregated view of this scenario, we proceed as follows. For each of the 30 models, we randomly select 50 users as a test set (not used for training the previous models). We then repeat the following 500 times: we select a random user among the 50 and select a random discriminative feature among four to set a random (uniform) value in it (belonging to the domain of each selected feature, for example, from 18 to 100 in the age feature). This creates a set of 15,000 fake profiles as inputs.

In scenario B, to obtain a more realistic scenario where profiles are created from real data from the dataset, we now proceed as follows. We also select 50 profiles from the dataset as a test set, so we can perform our core experiment: the four discriminatory features of each of these profiles are sequentially replaced by those of the 49 remaining profiles, and each resulting test profile is fed to the model for prediction. (This permits us to test the model with realistic values in these features, and the process creates 2,450 profiles to search for an IP.) We count the number of times the output risk label has switched, as compared to the original untouched profile fed to the model. We repeat this operation on the 30 models to observe deviations.

We note that as IPs can be found solely using the decisions provided by the classifier (see Definition 3), we do not need to rely on an explanation framework (such as, for example, LIME) in the experiments.

The low probability of findings IPs at random. In the case of scenario A, we compare the original label with the one obtained from each crafted input. Recall that a label change while considering two inputs constitutes an IP. We obtain 8.09% of IPs (standard deviation of 4.08).

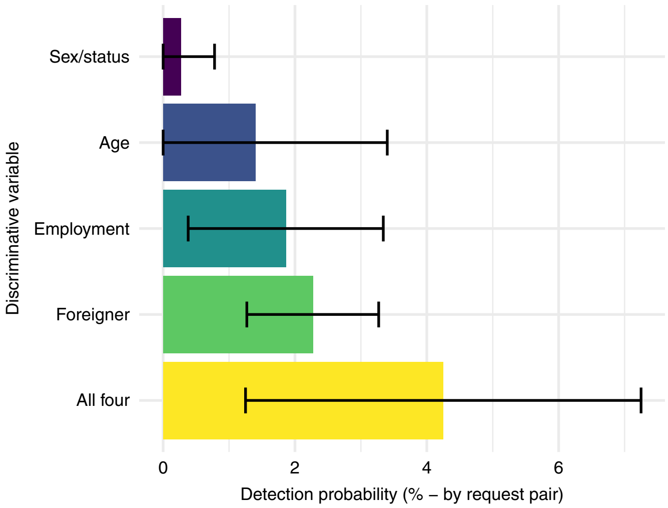

Fig. 4 | Percentage of label changes when swapping the discriminative features in the test set data for scenario B. Bars indicate standard deviations. These indicate the low probability to spot a PR attack on the provider model.

Figure 4 depicts, for scenario B, the proportion of label changes over the total number of test queries. If we change just one of the four features, we obtain, on average, 1.86%, 0.27%, 1.40% and 2.27% label changes (for employment, sex/status, age and foreigner features, respectively), while simultaneously changing four features gives a probability of 4.25%. The standard deviations are 1.48%, 0.51%, 1.65%, 2.17% and 3.13%, respectively.

This probability of 4.25% is higher than our deterministic lower bound $BP(|\mathcal{X}|)$ (Proposition 3), hinting that this discriminating classifier is easier to spot that the worst-case one. Moreover, because not finding an IP after some requests does not guarantee the absence of a discriminating behaviour, we now look at the user-side perspective: testing the absence of discrimination of a remote service. It turns out that we can compute an expectation of the number of queries for such a user to find an IP.

Users can query the service with inputs, until they are confident enough that such a pair does not exist. Assuming one seeks a 99% confidence level—that is, fewer than 1% of chances to falsely judge a discriminating classifier as non-discriminating—and using the detection probabilities of Fig. 4, we can compute the associated P values. A user testing a remote service based on these hypotheses would need to craft, respectively, 490, 2,555, 368, 301 and 160 pairs (for employment, sex/status, age, foreigner and all four, respectively) in the hope to decide on the existence or not of an IP, as presented in Fig. 5 (note the log scale on the y axis).

These experiments highlight the hardness of experimentally checking for PR attacks using intuitive approaches. Yet, we cannot claim that there are no efficient input space exploration strategies for finding IPs in a more practical way. This is certainly an interesting topic for future work.

Discussion

In this section we describe several consequences of the findings of this Article and some open questions.

Findings and applicability. We have shown that a malicious provider can always craft a fake explanation to hide its use of discriminatory features by creating a surrogate model for providing an

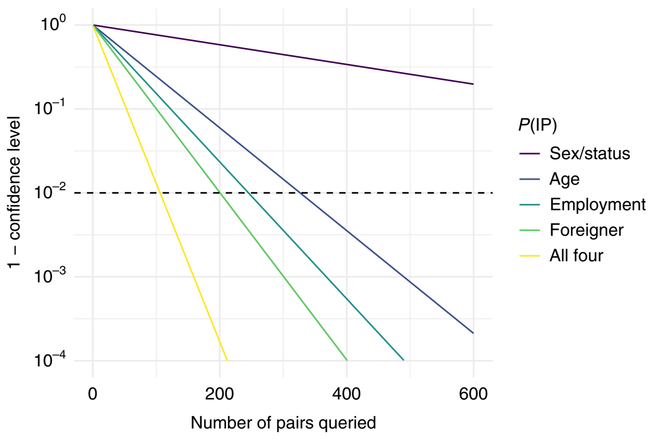

Fig. 5 | Confidence level as a function of the number of tested input pairs, based on the German Credit detection probability in Fig. 4. The dashed line represents the 99% confidence level.

explanation to a given user. An impossibility result follows, for a user to detect such an attack while using a single explanation. The detection by a user, or a group of users, is possible only in the case of multiple and deliberate queries ($BP(\epsilon > 1)$), and this process may require an exhaustive search of the input space.

However, we see that our practical experiment on the German Credit dataset is far from this complexity. Intuitively, the probability of finding an IP is proportional to the 'discrimination level' of a classifier. Although quantifying such a level is a difficult task, we explore a possible connection in the next section.

We note that the malicious providers have another advantage for covering PR attacks. Because multiple queries must be issued to spot inconsistencies via IP pairs, basic rate-limiting mechanisms for queries may block and ban the incriminated users. Defences of this kind, for preventing attacks on online machine services exposing APIs, are being proposed[29]. This adds another layer of complexity for the observation of misbehaviour.

Connection with disparate impact. We now briefly relate our problem to disparate impact. A recent article[30] proposed adopting 'a generalization of the 80 percent rule advocated by the US Equal Employment Opportunity Commission (EEOC)' as a criterion for disparate impact. This notion of disparate impact proposes to capture discrimination through the variation of outcomes of an algorithm under scrutiny when applied to different population groups.

More precisely, let α be the disparity ratio. The authors propose the following formula, here adapted to our notation[30]:

$$\alpha = \frac{\mathbb{P}(y|x_{\mathrm{d}} = 0)}{\mathbb{P}(y|x_{\mathrm{d}} = 1)}$$

where $X_{\mathrm{d}} = \{0, 1\}$ is the discriminative space reduced to a binary discriminatory variable. Their approach is to consider that if $\alpha < 0.8$ then the tested algorithm could be qualified as discriminative.

To connect disparate impact to our framework, we conduct the following strategy. Consider a classifier C having a disparate impact α, and producing a binary decision $C(x) \in \{0 = \text{bounce},\ 1 = \text{enter}\}$. We search for IPs as follows: first, pick $x \in X_{\mathrm{l}}$, a set of legit features. Then take $a = (x, x_{\mathrm{d}} = 0)$, representing the discriminated group, and $b = (x, x_{\mathrm{d}} = 1)$, representing the undiscriminated group. Then test C on both a and b: if $C(a) \neq C(b)$ then (a,b) is an IP. The probability $\mathbb{P}$ of finding an IP in this approach can be written as $\mathbb{P}(IP)$. Let A (resp. B) be the event 'a enters' (resp. 'b enters').

We can develop

$$\begin{aligned}
\mathbb{P}(IP) &= \mathbb{P}(C(a) \neq C(b)) \\
&= \mathbb{P}(A \cap \overline{B}) + \mathbb{P}(\overline{A} \cap B) \\
&= \mathbb{P}(A) - \mathbb{P}(A \cap B) + \mathbb{P}(B) - \mathbb{P}(A \cap B) \\
&= \mathbb{P}(B)(1 + \alpha) - 2\mathbb{P}(A \cap B),\ \text{because}\ \alpha = \mathbb{P}(\mathrm{A})/\mathbb{P}(\mathrm{B})
\end{aligned}$$

Using conditional probabilities, we have $\mathbb{P}(A \cap B) = \mathbb{P}(B|A).\mathbb{P}(A)$. Thus $\mathbb{P}(IP) = \mathbb{P}(B)(1 + \alpha - 2\alpha.\mathbb{P}(B|A))$. Given that the conditional probability $\mathbb{P}(B|A)$ is difficult to assess without further hypotheses on C, let us investigate two extreme scenarios:

- Independence: A and B are completely independent events, even though a and b share their legit features in X_{l}. This scenario, which is not very realistic, could model purely random decisions with respect to attributes from X_{d}. In this scenario $\mathbb{P}(B|A) = \mathbb{P}(B)$.
- Dependence: $A \Rightarrow B$: if a is selected, despite its membership to the discriminated group ($a = (x,0)$), then necessarily b must be selected, as it can only be 'better' from C's perspective. In this scenario $\mathbb{P}(B|A) = 1$.

Figure 6 represents the numerical evaluation of our two scenarios. First, it shows that the probability of finding an IP strongly depends on the probability of a success for the non-discriminated group $\mathbb{P}(B)$. Indeed, because the discriminated group has an even lower probability of success, a low success probability for the non-discriminated group implies frequent cases where both a and b are failures, which does not constitute an IP.

In the absence of disparate impact ($\alpha = 1$), both scenarios provide very different results: the independence scenario easily identifies IPs, which is coherent with the 'random' nature of the independence assumption. This underlines the unrealistic nature of the independence scenario in this context. With a high disparate impact, however (for example, $\alpha = 0.1$), the discriminated group has a high probability of failure. The probability of finding an IP is therefore very close to the simple probability of the non-discriminated group having a success $\mathbb{P}(B)$, regardless of the considered scenario.

The purely discriminative classifier presented in the section 'PR attacks on purely discriminative classifiers' also constitutes an extreme case with respect to disparate impact: $\alpha = \mathbb{P}(A|a = female)/\mathbb{P}(B|b = male) = 0$. In Fig. 6, the case $\alpha = 0$ is not represented, but it lies on the diagonal $\mathbb{P}(IP) = \mathbb{P}(B)$, regardless of the scenario. Because $\mathbb{P}(B) = 1$ (males always enter), we deduce $\mathbb{P}(IP) = 1$. In other words, testing any (male, female) couple spots the attack.

The dependence scenario nicely illustrates a natural connection: the higher the disparate impact, the higher the probability to find an IP. Although this only constitutes a thought experiment, we believe this highlights possible connections with standard discrimination measures and conveys the intuition that, in practice, the probability of finding IPs exposing a PR attack strongly depends on the intensity of the discrimination hidden by that PR attack.

Open problems for remote explainability. Regarding the test efficiency, it is common for fairness assessment tools to leverage testing. As the features that are considered discriminating are often precise[11,17], the test queries for fairness assessment can be targeted and some notions of efficiency in terms of the amount of requests can be derived. This may be done by sampling the feature space under scrutiny, for instance (as in the work by Galhotra and colleagues[11]).

Yet, it appears that with current applications such as social networks[13], users spend a considerable amount of time online, producing more and more data that turn into features and also are the basis for the generation of other meta-features. In that context, the full scope of features, discriminating or not, may not be clear to a user.

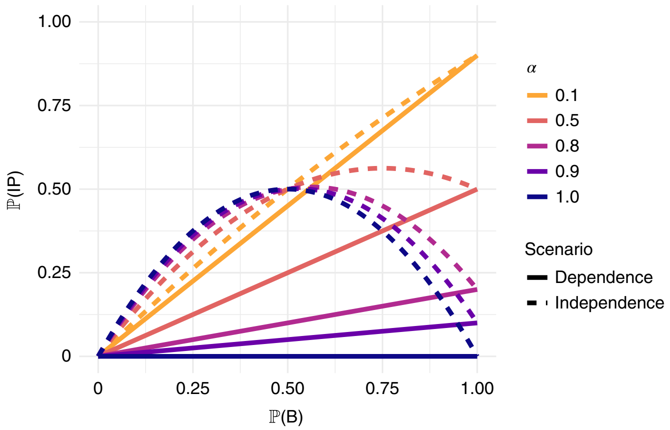

Fig. 6 | Probability to find an IP, as a function of $P(B)$, the probability of success for a non-discriminated group. Note that α represents the disparity ratio.

This makes exhaustive testing unreachable, even theoretically, due to the very likely non-complete picture of what providers are using to issue decisions. This is another challenge on the way to remote explainability—that providers are not willing to release a complete and precise list of all attributes leveraged in their system.

Another scenario is users sharing their observations for spotting discrimination, as is common in real life regarding bouncing issues. In practice, users can have a prior on which are discriminative inputs in specific applications and decide to coordinate for testing such a target application. By doing so, they collectively 'build' a surrogate model and coordinate its study, without ever relying on provided explanations. This approach bears some similarities with blackbox surrogate approaches, such as LIME (with the aim of finding IPs and for other benefits like fairness testing or model comparison). However, collecting uncoordinated user queries to span the entire input space is related to the coupon collector problem (requiring on the order of $|\mathcal{X}|\log(|\mathcal{X}|)$ independent user inputs). Although this solution might ultimately be the only one that holds, as in reality, its implementation is not straightforward as it raises privacy problems. In our example of the German Credit dataset, this would lead users to disclose sensitive data such as their savings or employment status.

Towards a provable explainability, some other computing applications, such as data storage or intensive processing, have also previously questioned the possibility of malicious service providers. Motivated by the plethora of offers in the cloud computing domain and the question of quality of service, protocols such as 'proof of data possession'[14] or 'proof-based verifiable computation'[31] assume that the service provider might be malicious. A solution to still have services executed remotely in this context is then to rely on cryptographic protocols to formally verify the work performed remotely. To the best of our knowledge, no such provable process logic has been adapted to explainability. This is certainly an interesting development to come.

Related work

Explaining in-house models. As a consequence of the major impact of machine learning models in many areas of our daily life, the notion of explainability has been pushed by policy makers and regulators. Many works address the explainability of inspected model decisions on a local set-up (see surveys in refs. [7,8,32])—some specifically for neural network models[33]—where the number of requests to the model is unbounded. Regarding the question of fairness, a recent work specifically targeted the fairness and discrimination of in-house software by developing a testing-based method[11].

Dealing with remote models. The case of models available through a remote blackbox interaction set-up is particular, as external observers are bound to scarce data (labels corresponding to inputs, while being limited in the number of queries to the blackbox[34]). Adapting the explainability reasoning to models available in a blackbox set-up is of major societal interest: Andreou and others[13] have shown that Facebook's explanations for their ad platform are incomplete and sometimes misleading. They also conjecture that malicious service providers can 'hide' the sensitive features used by explaining decisions with very common ones. In that sense, our Article is exposing the hardness of explainability in that set-up, confirming that malicious attacks are possible. Milli and others[35] provide a theoretical ground for reconstructing a remote model (a two-layer ReLu neural network) from its explanations and input gradients; if further research proves the approach practical for current applications, this technique may help to infer the use of discriminatory features by the service provider.

Operating without trust—the domain of security. In the domain of security and cryptography, some similar set-ups have found a large body of work to solve the trust problem. In proof of data possession protocols[14], a client executes a cryptographic protocol to verify the presence of her data on a remote server; the challenge that the storage provider responds to assesses the possession or not of some particular piece of data. Protocols can give certain or probabilistic guarantees. In proof-based verifiable computation[31], the provider returns the results of a queried computation, along with a proof for that computation. The client can then check that the computation indeed took place. These schemes, along with this Article exhibiting attacks on remote explainability, motivate the need for the design of secure protocols.

Discrimination and bias detection approaches. Our work is complementary to classic discrimination detection in automated systems. In contrast to works on fairness[36], which attempt to identify and measure discrimination from systems, our work does not aim to spot discrimination, as we have shown it can be hidden by the remote malicious provider. We instead are targeting the occurrence of incoherent explanations produced by such a provider with the intent to cover its behaviour, which has a completely different nature than fairness-based test suites. Galhotra and others[11], inspired by statistical causality[15], for example, propose to create input datasets for observing discrimination on some specific features by the system being tested.

Although there are numerous comments and proposals for good practice when releasing models that may include some forms of bias[37], the automatic detection of bias on the user side is also of interest for the community. For example, researchers have sought to detect Simpson's paradox[38] in the data[39]. Another work has made use of causal graphs to detect[40] a potential discrimination in the data, while ref. [41] proposes purging the data so that direct and/or indirect discriminatory decision rules are converted to legitimate classification rules. Some works are specific to some applications, such as financial ones[42]. Note that those approaches by definition require access to the training data, which is a too restrictive assumption in the context of our contribution.

The work in ref. [43] proposes leveraging transfer learning (or distillation) to mimic the behaviour of a blackbox model, here a credit scoring model. A collection campaign is assumed to provide a labelled dataset with risk scores, as produced by the model and ground-truth outcome. From this dataset a model is trained that aims at mimicking the blackbox as close as possible. Both models are then compared on their outcome, and a method to estimate the confidence interval for the variance of results is presented. The trained model can then be queried to assess potential bias. This approach proves solid guarantees when one assumes that the dataset is extracted from a blackbox that does not aim to bias its outputs to prevent audits of that form.

The rationalization of explanations. More closely related to our work is the recent paper by Aivodji and colleagues[16], which introduces the concept of rationalization, in which a blackbox algorithm is approximated by a surrogate model that is 'fairer' that the original blackbox. In our terminology, they craft C' models that optimize arbitrary fairness objectives. To achieve this, they explore decision tree models trained using the blackbox decisions on a predefined set of inputs. This produces another argument against blackbox explainability in a remote context. The main technical difference with our tree algorithm is that their surrogates C' optimize an exterior metric (fairness) at the cost of some coherence (fidelity in the authors' terminology). In contrast, our illustration produces surrogates with perfect coherence that do not optimize any exterior metric such as fairness. In our model, spotting an incoherence (that is, the explained model produces a y while the blackbox produces a $\bar{y}$) would directly provide a proof of manipulation and reveal the trickery. Interestingly, the IP approach fully applies in the context of their model surrogates, as it arises as soon as more than one surrogate is used (regardless of the explanation). This Article focuses on the user-side observation of explanations and users' ability to discover such attacks. We rigorously prove that single queries are not sufficient to determine a manipulation, and that the problem is hard even in the presence of multiple queries and observations.

Conclusion

In this Article, we have studied explainability in a remote context, which is sometimes presented as a way to satisfy society's demand for transparency when faced with automated decision making. We prove that it is unwise to blindly trust these explanations: like humans, algorithms can easily hide the true motivations of a decision when asked for explanation. To illustrate, we have presented an attack that generates explanations to hide the use of an arbitrary set of features by a classifier. Although this construction applies to any classifier queried in a remote context, we have also presented a concrete implementation of that attack on decision trees. On the defensive side, we have shown that such a manipulation cannot be spotted by one-shot requests, which is unfortunately the nominal use case. However, the proof of such trickery (pairs of decisions that are not coherent) necessarily exists. We have further evaluated in a practical scenario the probability of finding such pairs, which is low. It is thus arguably impractical for the attack to be detect by an isolated user.

We conclude that this must consequently question the whole concept of the explainability of a remote model operated by a third-party provider, at the very least. One research direction is to develop secure schemes in which the involved parties can trust the information exchanged about decisions and their explainability, as enforced by new protocols. A second line of research may be the collaboration of users' observations for spotting the attack in an automated way. Indeed, as was done by Chen and colleagues[44] to understand the impact of Uber surge pricing on passengers and drivers (by deploying 43 Uber accounts that act as clients), data can be shared to retrieve information more reliably. The anonymization of users' data if a pool of measurements is to be made public is certainly crucial to ensure scalable observations of blackbox decision-making systems. We believe that this is an interesting development to come in relation to the promises of AI and automated decision-making processes.

Data availability

The data that support the findings in this study—as the German Credit dataset—are publicly available at https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data).

Code availability

The code used for the experiments is provided at https://github.com/erwanlemerrer/bouncer_problem (https://doi.org/10.5281/zenodo.3907271).

Received: 29 November 2019; Accepted: 10 July 2020;
Published online: 24 August 2020

References

1. Veale, M. Logics and practices of transparency and opacity in real-world applications of public sector machine learning. In *Proceedings of the 4th Workshop on Fairness, Accountability and Transparency in Machine Learning* (FAT/ML, 2017); https://arxiv.org/pdf/1706.09249.pdf
2. de Laat, P. B. Algorithmic decision-making based on machine learning from big data: can transparency restore accountability? *Philos. Technol.* **31**, 525–541 (2018).
3. Naumov, M., et al. Deep learning recommendation model for personalization and recommendation systems. Preprint at https://arxiv.org/pdf/1906.00091.pdf (2019).
4. Goodman, B. & Flaxman, S. European Union regulations on algorithmic decision-making and a 'right to explanation'. *AI Magazine* **38**, 50–57 (2017).
5. Selbst, A. D. & Powles, J. Meaningful information and the right to explanation. *International Data Privacy Law* **7**, 233–242 (2017).
6. Adadi, A. & Berrada, M. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE Access* **6**, 52138–52160 (2018).
7. Guidotti, R. et al. A survey of methods for explaining black box models. *ACM Comput. Surveys* **51**, 93 (2018).
8. Molnar, C. *Interpretable Machine Learning* (GitHub, 2019); https://christophm.github.io/interpretable-ml-book/
9. Zhang, Y. & Chen, X. Explainable recommendation: a survey and new perspectives. Preprint at https://arxiv.org/pdf/1804.11192.pdf (2018).
10. Ribeiro, M. T., Singh, S. & Guestrin, C. 'Why should I trust you?': explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 1135–1144 (ACM, 2016); https://doi.org/10.1145/2939672.2939778
11. Galhotra, S., Brun, Y. & Meliou, A. Fairness testing: testing software for discrimination. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* 498–510 (ESEC/FSE, 2017); https://doi.org/10.1145/3106237.3106277
12. Lundberg, S. M. & Lee, S.-I. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* 4768–4777 (NIPS, 2017).
13. Andreou, A. et al. *Investigating Ad Transparency Mechanisms in Social Media: A Case Study of Facebook's Explanations* (NDSS, 2018); https://doi.org/10.14722/ndss.2018.23204
14. Ateniese, G. et al. Provable data possession at untrusted stores. In *Proceedings of the 14th ACM Conference on Computer and Communications Security* 598–609 (ACM, 2007); https://doi.org/10.1145/1315245.1315318
15. Pearl, J. Causal inference in statistics: an overview. *Stat. Surveys* **3**, 96–146 (2009).
16. Aivodji, U. et al. Fairwashing: the risk of rationalization. In *Proceedings of the 36th International Conference on Machine Learning* (eds Chaudhuri, K. & Salakhutdinov, R.) 161–170 (PMLR, 2019).
17. Hajian, S., Domingo-Ferrer, J. & Martínez-Ballesté, A. Rule protection for indirect discrimination prevention in data mining. In *Modeling Decision for Artificial Intelligence* (eds Torra, V., Narakawa, Y., Yin, J. & Long, J.) 211–222 (Springer, 2011).
18. Menon, A. K. & Williamson, R. C. The cost of fairness in binary classification. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency* (eds Friedler, S. A. & Wilson, C.) 107–118 (PMLR, 2018).
19. Tramèr, F., Zhang, F., Juels, A., Reiter, M. K. & Ristenpart, T. Stealing machine learning models via prediction APIs. In *Proceedings of the 25th USENIX Conference on Security Symposium, SEC'16* 601–618 (USENIX Association, 2016).
20. Miller, T. Explanation in artificial intelligence: insights from the social sciences. Preprint at https://arxiv.org/pdf/1706.07269.pdf (2017).
21. Cummins, D. D., Lubart, T. & Alksnis, O. Conditional reasoning and causation. *Memory Cognition* **19**, 274–282 (1991).
22. Alexander, L. What makes wrongful discrimination wrong? Biases, preferences, stereotypes and proxies. *University of Pennsylvania Law Review* **141**, 149–219 (1992).
23. Wu, X. et al. Top 10 algorithms in data mining. *Knowledge Inform. Syst.* **14**, 1–37 (2008).
24. Quinlan, J. R. *C4.5: Programs for Machine Learning* (Elsevier, 2014).
25. *Statlog (German Credit Data) Data Set* (UCI, accessed 1 September 2019); https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)
26. Oreski, S. & Oreski, G. Genetic algorithm-based heuristic for feature selection in credit risk assessment. *Expert Syst. Appl.* **41**, 2052–2064 (2014).

27. Brock, A., Donahue, J. & Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. Preprint at https://arxiv.org/pdf/1809.11096.pdf (2019).
28. Khashman, A. Neural networks for credit risk evaluation: investigation of different neural models and learning schemes. *Expert Syst. Appl.* **37**, 6233–6239 (2010).
29. Hou, J. et al. Ml defense: against prediction API threats in cloud-based machine learning service. In *Proceedings of the International Symposium on Quality of Service*, *IWQoS '19* 7:1–7:10 (ACM, 2019)
30. Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C. & Venkatasubramanian, S. Certifying and removing disparate impact. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 259–268 (ACM, 2015); https://doi.org/10.1145/2783258.2783311
31. Braun, B. et al. Verifying computations with state. In *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles* 341–357 (ACM, 2013); https://doi.org/10.1145/2517349.25227332013
32. Datta, A., Sen, S. & Zick, Y. Algorithmic transparency via quantitative input influence: theory and experiments with learning systems. In *Proceedings of the 2016 IEEE Symposium on Security and Privacy* (*SP*) 598–617 (IEEE, 2016).
33. Yeh, C.-K., Kim, J., Yen, I. E.-H. & Ravikumar, P. K. Representer point selection for explaining deep neural networks. In *Proceedings of Advances in Neural Information Processing Systems 31* (eds Bengio, S. et al.) 9291–9301 (Curran Associates, 2018).
34. Tramèr, F., Zhang, F., Juels, A., Reiter, M. K. & Ristenpart, T. Stealing machine learning models via prediction APIs. In *Proceedings of the 25th USENIX Security Symposium* (*USENIX Security 16*) 601–618 (USENIX Association, 2016).
35. Milli, S., Schmidt, L., Dragan, A. D. & Hardt, M. Model reconstruction from model explanations. In *Proceedings of the Conference on Fairness, Accountability and Transparency*, *FAT* '19* 1–9 (ACM, 2019).
36. Binns, R. Fairness in machine learning: lessons from political philosophy. In *Proceedings of the 2018 Conference on Fairness, Accountability and Transparency* Vol. 81, 149–159 (PMLR, 2017).
37. Mitchell, M. et al. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability and Transparency*, *FAT* '19* 220–229 (ACM, 2019).
38. Blyth, C. R. On Simpson's paradox and the sure-thing principle. *J. Am. Stat. Assoc.* **67**, 364–366 (1972).
39. Alipourfard, N., Fennell, P. G. & Lerman, K. Using Simpson's paradox to discover interesting patterns in behavioral data. Preprint at https://arxiv.org/pdf/1805.03094.pdf (2018).
40. Zhang, L., Wu, Y. & Wu, X. Achieving non-discrimination in data release. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *KDD '17* 1335–1344 (ACM, 2017).
41. Hajian, S. & Domingo-Ferrer, J. A methodology for direct and indirect discrimination prevention in data mining. *IEEE Trans. Knowledge Data Eng.* **25**, 1445–1459 (2013).
42. Zhang, Y. & Zhou, L. Fairness assessment for artificial intelligence in financial industry. Preprint at https://arxiv.org/pdf/1912.07211.pdf (2019).
43. Tan, S., Caruana, R., Hooker, G. & Lou, Y. Distill-and-compare: auditing black-box models using transparent model distillation. In *Proceedings of the 2018 AAAI/ACM Conference* 303–310 *AIES* (AAAI, 2018); https://doi.org/10.1145/3278721.3278725
44. Chen, L., Mislove, A. & Wilson, C. Peeking beneath the hood of Uber. In *Proceedings of the 2015 Internet Measurement Conference*, *IMC '15* 495–508 (ACM, 2015).

Author contributions

The theoretical framework was developed by E.L.M. and G.T. Experimental work was carried out by E.L.M. and data analysis by G.T.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to E.L. or G.T.

Reprints and permissions information is available at www.nature.com/reprints.