Collaborative Explanation of Deep Models with Limited Interaction for Trade Secret and Privacy Preservation

Josep Domingo-Ferrer
Universitat Rovira i Virgili
Dept. of Computer Science and Mathematics
UNESCO Chair in Data Privacy
CYBERCAT-Center for Cybersecurity Research of Catalonia
Tarragona, Catalonia
josep.domingo@urv.cat

Cristina Pérez-Solà
Universitat Rovira i Virgili
Dept. of Computer Science and Mathematics
UNESCO Chair in Data Privacy
CYBERCAT-Center for Cybersecurity Research of Catalonia
Tarragona, Catalonia
cristina.perez@urv.cat

Alberto Blanco-Justicia
Universitat Rovira i Virgili
Dept. of Computer Science and Mathematics
UNESCO Chair in Data Privacy
CYBERCAT-Center for Cybersecurity Research of Catalonia
Tarragona, Catalonia
alberto.blanco@urv.cat

ABSTRACT

An ever increasing number of decisions affecting our lives are made by algorithms. For this reason, algorithmic transparency is becoming a pressing need: automated decisions should be explainable and unbiased. A straightforward solution is to make the decision algorithms open-source, so that everyone can verify them and reproduce their outcome. However, in many situations, the source code or the training data of algorithms cannot be published for industrial or intellectual property reasons, as they are the result of long and costly experience (*e.g.* this is typically the case in banking or insurance). We present an approach whereby individual subjects on whom automated decisions are made can elicit in a *collaborative* and *privacy-preserving* manner a rule-based approximation of the model underlying the decision algorithm, based on limited interaction with the algorithm or even only on how they have been classified. Furthermore, being rule-based, the approximation thus obtained can be used to detect potential discrimination. We present empirical work to demonstrate the practicality of our ideas.

CCS CONCEPTS

• **Software and its engineering** → **Empirical software validation**; • **Information systems** → *Collaborative and social computing systems and tools*; Data analytics; • **Social and professional topics** → *Automation*; *Codes of ethics*; *Technology audits*; • **Security and privacy** → Privacy-preserving protocols; • **Computing methodologies** → Machine learning.

KEYWORDS

Machine Learning, Transparency, Explainability, Auditing, Privacy

ACM Reference Format:
Josep Domingo-Ferrer, Cristina Pérez-Solà, and Alberto Blanco-Justicia. 2019. Collaborative Explanation of Deep Models with Limited Interaction for Trade Secret and Privacy Preservation. In *Workshop on Fairness, Accountability, Transparency, Ethics, and Society on the Web (FATES'19), May 2019, San Francisco, California USA.* ACM, New York, NY, USA, 7 pages.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
FATES'19, May 2019, San Francisco, California USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6675-5/19/05…$15.00

1 INTRODUCTION

Recent advances in machine learning and artificial intelligence, tightly coupled with the great availability of data, are driving companies towards automated decisions. Computers can make decisions much faster than their human counterparts while taking many more variables into account. Artificial neural networks and support vector machines are prevalent machine learning techniques in current decision-making tools, which results in a major drawback: the inability or at any rate the difficulty to explain why a certain decision was made on certain input data. Explaining machine learning models is of great interest both for technical, legal and ethical reasons. On the technical side, explainable machine learning allows data scientists to fine-tune their models to achieve better performance. On the legal and ethical sides, new legislation such as the EU General Data Protection Regulation [4] and recommendations like the European Commission's Ethics Guidelines for Trustworthy AI [9] urge organizations making automated decisions to be ready to explain them at the request of the affected people.

A data scientist can certainly get some insight on a particular decision by looking at the activations of a neural network, or even about the whole model by visualizing the weights in convolutional layers, but this information is usually of little to no interest to the layman. Older machine learning models based on rules were much more expressive in this regard, because just looking at the rules triggered to reach a certain decision was an explanation in itself. However, rule-based models have been clearly outperformed by (the less explainable) deep models.

Several works, such as [13–16, 19], propose mechanisms to equip deep models with explainability. A usual approach is to treat the deep model as a black box and use the decisions of the model to build simpler or more localized models that are easier to explain. These explainability proposals, however, make assumptions that limit their use, again, to data scientists rather than the general public. Typical assumptions are knowledge of the whole training data set or unrestricted access to the deep models (even if these are treated as black boxes). An individual who has been affected by an automated decision is very unlikely to have access to the training data or to the deep model:

- On the one hand, deep models require large amounts of data to be trained, which normally come either from an already existing service by the same organization offering the machine learning service or from a data market. Thus, in most cases these data sets cannot be made available to the public because, first, they are a highly valued asset of the organization that has collected the data and, second, they may contain personally identifiable information, and thus be subject to privacy laws.
- On the other hand, the training process of deep models needs to be carefully fine-tuned, from the architecture of the model to meta-parameters. This fine tuning takes time and computing power (and therefore money), which makes a well-trained deep model another highly valued asset that organizations most probably wish to protect as a trade secret.

Beyond explainability, non-discrimination is also a societal requirement on automated decisions. Decision-making tools should not discriminate people belonging to disadvantaged or vulnerable groups as established in the Universal Declaration of Human Rights. Therefore, deep models should not produce biased decisions based on gender, ethnicity, sexual orientation, religious beliefs, etc. If decisions can be explained in terms of rules, then anti-discrimination frameworks such as those proposed by Hajian *et al.* [6–8] can be used to detect discriminatory rules and sanitize the training data set so that the resulting rules be non-discriminatory.

Contribution and plan of this paper

In this paper, we present a methodology that allows individual subjects on whom automated decisions have been made to elicit in a collaborative and privacy-preserving manner a rule-based approximation of the model underlying the decision algorithm. Individuals can do so based on limited interaction with the algorithm or even only on how they have been classified. Furthermore, being rule-based, the approximation thus obtained allows detecting potential discrimination. Overall, the proposed system provides a way to empower users against opaque decision-making systems. Therefore, users are incentivized to collaborate for their own benefit, which in turn benefits the entire users' community: this is known as a co-utile situation [2].

We base our proposed methodology on building explanatory rule-based models for specific queries to a deep model. While this is not entirely new, the novelty of our proposal is that we assume that service providers limit the number of queries a single user can make, in order to preserve their deep model as a trade secret. This limitation makes it impractical for a single user to probe the model. To overcome this problem, we propose two collaborative methods:

- One method is predicated on the availability of a simulator of the deep model (such as those offered by insurance companies to potential customers to calculate fees based on personal details). Each user can query this simulator a limited number of times. By agreeing on the queries to be made by each user, a collaborating community of users can cover the entire domain of input attributes and thereby build a general explanatory approximation of the deep model. Sharing the input features and output labels of the queries does not pose privacy problems because in general users do not query with their own true input features.
- On the other hand, we explore a more "frugal" method for the case in which no simulator is available and each user is only allowed to query the deep model once, in principle with her own features. To build the model approximation, users need to share their input features and the labels returned by their queries. This poses privacy problems if users share their true features and accuracy problems if they do not. We propose solutions to deal with this privacy-accuracy trade-off.

Section 2 gives background on explainability, anti-discrimination sanitization and the conflict with trade secret. Section 3 describes the collaborative rule-based model approximation method with a simulator and Section 4 describes the method without a simulator. Section 5 presents empirical results on the performance of both methods. Finally, conclusions and future research lines are gathered in Section 6.

2 BACKGROUND

2.1 Explainability in machine learning

Article 22 of the General Data Protection Regulation (GDPR, [4]) states that citizens may not be subjected to decisions based only on automated mechanisms, and may request an explanation of such decisions [5]. It has however been noted in [21] that the explainability requirement in the GDPR is not clear enough. Nonetheless, other documents by the European Commission, such as the Ethics Guidelines for Trustworthy AI [9], distinctly emphasize the need for explainability. Unfortunately, explaining currently employed machine learning models is a highly complicated matter, since their complexity makes it extremely difficult to interpret their decisions.

The arduousness of explaining deep models is unsettling not only for the individuals affected by their decisions, but also for the artificial intelligence practitioners themselves. If a model makes mistakes, being able to interpret its wrong decisions would greatly help AI engineers to tune certain parameters and/or improve the curation and processing of the training data sets.

Several methods have been proposed to generate explanations for machine learning algorithms whose internal details are undisclosed. Some of these methods are focused on explaining decisions made by a particular model, *e.g.* a neural network [16] or a support vector machine [13], whereas others are model-agnostic [14, 15, 19] and allow explaining decisions made by any classification algorithm without being aware of its underlying model.

Machine learning explainability methods can also be classified according to the explanatory models they use. Some proposals provide visual and textual explanations, like [10]. Yet, the most common explanatory models are decision rules [15, 19], decision trees [17], and linear models [18]. For any of these explanatory models to be useful for human interpretation, the complexity of the explanations must be low enough. For instance, a two-level decision tree may be interpretable for humans, whereas a hundred-level decision tree is unlikely to serve the purpose.

Yet another categorization is between local and global explanatory methods. Some solutions generate local explanatory models that approximate the hidden (deep) model in the vicinity of a given point [13–15, 19]. In contrast, other solutions aim at explaining the

entire model [20]. In either case, the goal is for the explanatory models to predict the outcome of the hidden models as accurately as possible.

We next review in slightly more detail two of the above-mentioned proposals, that are model-agnostic and local. LIME [14] provides local explanations of decisions by sampling instances around the provided feature set (*i.e.* the user query) and interrogating the model to obtain the corresponding labels. This set is then used to train an explainable local model, such as decision trees. The authors test whether the explanations are faithful to the original model, whether the explanations increase the trust in the predictions and the model, and whether human non-experts can choose the best models and improve them. The same authors later proposed a method dubbed Anchors [15], in which local explanations of predictions are provided as rules (called anchors). An anchor is defined as a decision rule that has as precision (with respect to the hidden model) over a threshold. For a given prediction, different anchors may exist, and the authors propose algorithms to search for the one with the highest coverage, that is, the one that matches the largest amount of input samples. The paper also provides experimental evaluation of the ability of anchors to predict new unseen instances, both in terms of the amount of predictions that can be performed with a set of anchors computed from other inputs and the precision of those predictions with respect to the hidden model.

2.2 Discrimination detection and correction

Being able to explain decision results is a major asset to identify and correct discrimination or bias in machine learning models. While an automated classifier may be regarded as a fair decision-making tool, it is not necessarily so. In particular, if the classifier was trained on inherently biased data, the model it learned is likely to result in discriminatory decisions. Works by Hajian *et al.* [6–8] use rule-based models to tackle discrimination detection and correction, either by identifying and pre-processing inherently discriminatory instances in the training data, or by directly acting on the mined rules by eliminating some of them and/or generalizing some of their conditions. Clearly, these strategies are intended for explainable models, because changing the mined rules in a utility-preserving way requires understanding them. In [23], the authors focus on a different kind of discrimination, namely the different misclassification rates in different groups of people. The authors argue that a classifier should perform similarly for all instances, and propose a methodology that requires access to the ground truth (*i.e.* the training data).

2.3 Conflict between explainability, anti-discrimination and trade secret

All the above approaches to explainability and anti-discrimination either assume complete knowledge of the training data set or unlimited access by the users to the decision-making tool (although [15] explicitly tries to limit the amount of queries). We argue that these assumptions are unrealistic in production environments where the user is the subject affected by the decisions, rather the designer of the decision-making tool. The reason is that, as discussed in Section 1, both the training data and the decision model are usually protected by the service provider as trade secrets.

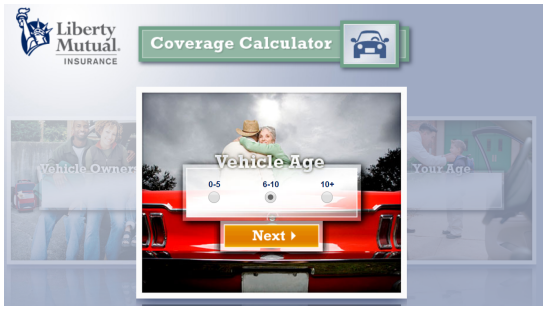


Figure 1: Simulator to calculate car insurance premiums

3 COLLABORATIVE RULE-BASED MODEL APPROXIMATION WITH A SIMULATOR

Although the internals of corporate decision-making algorithms are often hidden to the public, in some scenarios the service providers publish tools that allow querying their models *in a limited or controlled way*. The internals of the model remain hidden, but users are able to obtain responses from the algorithm for *a few* feature sets. Take as examples the web-based simulators for mortgage loans or car insurances that some banks or insurers make available on their websites[1]. Prospective customers can fill in their data and get a quotation for the monthly cost of a mortgage or the premium of an insurance. Figure 1 shows one such simulator.

Using public simulators to query a hidden model has one immediate consequence. Since users fill in the input data for the simulator themselves, the inputs may not correspond to the real user data (users can modify their inputs to observe how predictions change, or they can even lie). Therefore, they can query the model for feature sets that do not necessarily describe them. This is usually not possible whenever the evaluation is performed directly by the service provider. For instance, bank employees request proofs of income, work contracts, or official employment records in order to extract the feature sets to be used to make decisions on mortgages. Car insurance agents request the prospective customer's driving license and check previous insurance policies and reported accidents in their databases before offering a premium quotation.

To protect their model against deceptive users probing it, service providers often restrict the number of queries that any single user can submit to the simulator. This restriction is enforced, for example, by requiring the user to report a valid phone number, that is verified by the service provider before offering any results on the simulation. The simulator can also enforce throttling mechanisms in their APIs, which will limit he number of queries that any single user, or all users in general, can perform per unit of time. Other limiting strategies include the use of *CAPTCHAs* or any other anti-bot mechanism.

In this scenario, where a limited number of queries on arbitrary feature sets can be submitted by each user to a public simulator, user collaboration is a natural strategy to approximate the hidden model and understand the decisions it is making. This brings many benefits to the users and to society: (1) discriminatory behavior

[1] https://welcome.libertymutual.com/insurance-coverage-calculator-estimator/auto.html

can be detected, (2) inconsistencies in the predictions given by the model can be discovered, and (3) users unable to query the model themselves can learn the answer the model would output for their feature set.

3.1 Decision rule elicitation

Since users can query the simulator to obtain the answers to some feature sets, the most immediate technique they can use to collaboratively approximate the hidden deep model is to publish those answers. However, a more concise representation of the simulator's answers is needed to obtain human-understandable explanations of the hidden deep model.

Therefore, in addition to publishing the individual predictions each user learns from the simulator, we propose to mine *decision rules* from the pooled predictions. These rules have the advantage of explaining the model predictions and, at the same time, they are concise and they describe the predictions of the simulator for many possible feature sets.

A decision rule is an expression $X \rightarrow C$, where X is a a set of features (the premise or body of the rule) and C is a classification label. The *support* of the set of features X, $supp(X)$ is the number of predictions that take X as part of their input. Similarly, the support of a rule, $supp(X, C)$, (also sometimes called the coverage of the rule) is the number of predictions that contain X and C. The *confidence* of a decision rule, $conf(X \rightarrow C)$, measures how often C appears in predictions that contain X; hence, if $supp(X) > 0$, we have $conf(X \rightarrow C) = supp(X, C)/supp(X)$. Finally, a *frequent* decision rule is one with a support greater than a certain threshold τ_s and with a confidence greater than a certain threshold τ_c.

We propose to use a collaborative procedure generalizing the approach of Anchors [15] for a community of users to elicit decision rules that explain the predictions they receive. Specifically, users follow Protocol 1:

Protocol 1 (Rule elicitation with a simulator).

(1) *Each user in the community does:*
   (a) *If no predictions have been published so far by other users in the community, choose a region of feature sets, not necessarily containing the user's true features (*e.g. *if features are Age and Zipcode, choose an age range and a zipcode range). On the contrary, if some predictions have been already published, choose a feature set region that contains no published predictions.*
   (b) *Query the simulator as many times as allowed by randomly picking feature sets in the chosen region. As a result, a prediction for each feature set will be obtained.*
   (c) *Publish the predictions, where each prediction includes the input feature set and the simulator's answer.*
(2) *Any user can do:*
   (a) *Mine frequent decision rules from the published predictions.*
   (b) *Publish the mined decision rules.*

As mentioned in Section 1, a user incurs no privacy risks when sharing the predictions she has obtained from the simulator, because in general the input feature sets are not hers.

3.2 Detecting discrimation

Previous works have focused on providing metrics for quantifying the degree of discrimination of a decision rule [6, 11, 12]. A commonly employed metric is the extended lift (*elift*), that quantifies the degree of direct discrimination of a rule $A, B \rightarrow C$ whose premise contains a subset of features A defining a protected/vulnerable group (*e.g.* A could contain `Gender=woman` and/or `Religion=muslim`) along with other features B. Then

$$elift(A, B \rightarrow C) = \frac{conf(A, B \rightarrow C)}{conf(B \rightarrow C)}.$$

The intuition behind *elift* is to measure the effect of belonging to the protected group on the decision. There is no discrimination if and only if *elift* is 1, in which case belonging to the protected group is irrelevant to the decision.

The predictions and rules published as a result of Protocol 1 can be used to detect discriminatory behaviors that may exist in the hidden model. This can be done by *any user in the community* by running Algorithm 1, where we consider that a rule is discriminatory if its *elift* is greater than a threshold α:

Algorithm 1 (Detection of discrimination(α)).

(1) *For each published rule $A, B \rightarrow C$ whose premise contains a subset of features A defining a protected group, use the published predictions to compute $elift(A, B \rightarrow C)$.*
(2) *Return as discriminatory the rules whose elift is greater than α.*

If Algorithm 1 returns a non-empty set of discriminatory rules, the user running the algorithm can complain to the service provider, who should endeavor to correct that discrimination using the techniques recalled in Section 2.2.

3.3 Malicious actors

The system formed by Protocol 1 and Algorithm 1 can be easily extended to account for malicious actors. On the one hand, a malicious user may want to publish fake rules, that do not describe the behavior of the target hidden deep model, with the goal of undermining the credibility of the service provider. On the other hand, a malicious service provider may be hiding some of the features that are being used by the algorithm, and thus may generate different predictions for what seems to be the same data point from the users' perspective.

Although it is difficult to prevent both types of malicious behavior altogether, the following two minor modifications to Protocol 1 allow detecting cheaters:

- At Step 1b of Protocol 1, we require the simulator to return its responses digitally signed by the service provider. Each signed response includes the input feature set and the label returned by the simulator.
- At Step 1c the user publishes the predictions signed by the service provider.

With the above modifications, if the service provider's simulator returns two different predictions for the same feature set, there exists a proof of its misbehavior, as both predictions bear the service provider's signature. Also, users cannot inject fake predictions, because they would not carry the service provider's signature.

On the other hand, the process of mining decision rules is reproducible so that any fake rules published by malicious users can be falsified by anyone based on the signed predictions. In this way, fake rules can be discarded.

The downside of making the modifications described in this section is that they imply an additional overhead, both in terms of computational cost (generating and validating the signatures) and storage (each published prediction is appended a digital signature).

4 COLLABORATIVE RULE-BASED MODEL APPROXIMATION WITHOUT A SIMULATOR

In this scenario, the service provider does not supply any particular tool to test the deep model and obtain local explanations, so all a particular user can do is to query the deep model once, presumably to obtain the prediction/decision on her features.

If a community of users consists of individuals whose features are diverse enough, users could pool the predictions received by each of them and then engage in rule elicitation in the way described in Step 2 of Protocol 1. That would yield a set of rules offering an approximate explanation of the hidden model. These rules could also be examined to detect discrimination as per Algorithm 1.

The problem with the above approach is that if users share the predictions they receive based on their true features, they are revealing personal information, whereas if they use completely fake features a utility loss is likely. Hence, to be able to gather the large numbers of predictions required to approximate the model, we need to alleviate the privacy leakage in a clever way. We will resort to randomized response (RR, [1, 22]) to enable users to "lie" about the value of their features and/or classification label —that is, a user will be able to plausibly deny that the values she has reported are her true values— but do so in a "controlled" way —that is, in a way that still allows reconstructing the *true joint distribution* of features and classification labels.

Consider $\mathbf{X}$ to be a feature attribute that can take r possible values. Then the randomized response $\mathbf{Y}$ reported by the user instead of $\mathbf{X}$ is computed using

$$\mathbf{P} = \begin{pmatrix} p_{11} & \cdots & p_{1r} \\ \vdots & \vdots & \vdots \\ p_{r1} & \cdots & p_{rr} \end{pmatrix}$$

where $p_{uv} = \Pr(\mathbf{Y} = v|\mathbf{X} = u)$, for $u, v \in \{1, \ldots, r\}$ denotes the probability that the randomized response is v when the respondent's true attribute value is u.

A strong point of RR is that, while it allows plausible deniability to users, it also allows estimating the proportion of true responses based on the reported responses. Let $\pi_1, \ldots, \pi_r$ be the proportions of users whose true values fall in each of the r categories of $\mathbf{X}$; let $\lambda_v = \sum_{u=1}^{r} p_{uv}\pi_u$ for $v = 1, \ldots, r$, be the probability of the reported value $\mathbf{Y}$ being v. If we define $\lambda = (\lambda_1, \ldots, \lambda_r)^T$ and $\pi = (\pi_1, \ldots, \pi_r)^T$, then $\lambda = \mathbf{P}^T\pi$. Furthermore, if $\hat{\lambda}$ is the vector of sample proportions corresponding to λ and $\mathbf{P}$ is nonsingular, it is proven in [1] that an unbiased estimator of π can be obtained as

$$\hat{\pi} = (\mathbf{P}^T)^{-1}\hat{\lambda}. \quad (1)$$

However, we are not interested in Expression (1) estimating the distribution of a single feature. Rather, we want that expression to estimate the true joint empirical distribution of all features and the classification label. From that joint empirical distribution, it will be possible for users to elicit rules approximately explaining the model. As pointed out in [3], to estimate the joint distribution, we need to consider as $\mathbf{X}$ the *Cartesian product of all features attributes and the classification label*. On the other hand, the probability matrix $\mathbf{P}$ should be agreed upon by all the community of users. Thus we have the following rule elicitation protocol:

Protocol 2 (Rule elicitation without a simulator ($\mathbf{P}$)).

(1) *Each user in the community does:*
   (a) *Query the service provider's model with her own features.*
   (b) *Use RR with matrix* $\mathbf{P}$ *to randomize the vector of her features and the classification label.*
   (c) *Publish her randomized vector.*
(2) *Any user can do:*
   (a) *Compute the empirical joint distribution of the collection of published randomized prediction vectors.*
   (b) *Use Expression (1) on the computed empirical joint distribution to estimate the true empirical joint distribution of the features and the classification label.*
   (c) *Re-create a synthetic data set by repeating each combination of features and the classification label as many times as indicated by the joint distribution.*
   (d) *Mine frequent decision rules from the synthetic data set.*
   (e) *Publish the mined decision rules.*

5 EMPIRICAL RESULTS

In this section we present experiments based on the Adult dataset. We view the data set as the result of collaborative users pooling the responses they have obtained from a black-box model after submitting their individual queries.

5.1 Results of the method with a simulator

We replicated the experiments of the Anchors paper [15] for the Adult data set using a neural network as the hidden model. Adult contains several demographic feature attributes and a classification attribute, predicting whether an individual makes more or less than 50K$/year. That is, we trained a neural network with 80% of the samples in the Adult data set and we used the rest of the data set as test samples to compare the trained model and its rule-based approximate explanation. Specifically, we created rules to explain the predictions of the model for half of the test samples. This would correspond to the rules that users elicit by querying the simulator repeatedly and then publish to approximate the hidden deep model. Then, we used the second half of the test samples as validation data set to test how good were the extracted rules at predicting the behavior of the model. This would correspond to feature sets of other users that do not query the model themselves, but rather use the previously published rule-based approximation to obtain an estimate of the way the hidden model would classify them.

Results on rule support/coverage and precision are already reported in the original Anchors paper, and are thus omitted here. Rather, we analyze conflicting rules, that is, how often two different

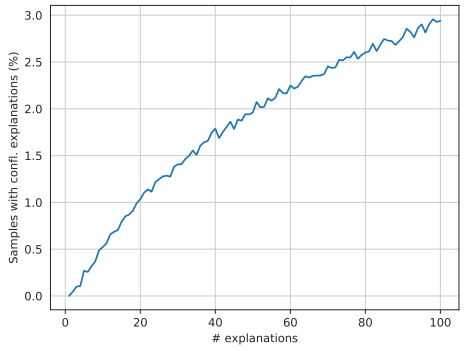


Figure 2: Percentage of samples with conflicting rules

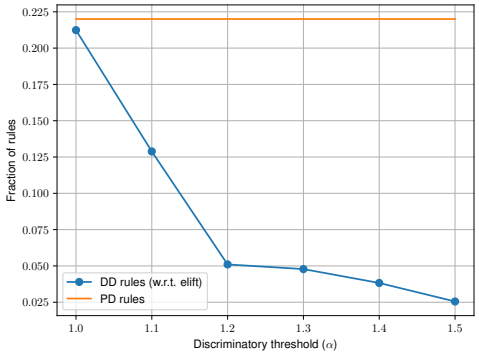


Figure 3: Fraction of potentially discriminatory rules (PD) and α-discriminatory rules (DD) with respect to $elift$.

rules are applicable to the same sample but provide contradictory predictions. Figure 2 shows the percentage of samples in the validation data set for which conflicting rules exist, as the number of published rules increases.

We have also analyzed how many of the elicited rules are discriminatory with respect to the *elift* metric. We have considered the subset of features defining a protected group to be A = {Sex = Female, Age $\leq$ 28}. There are 22% potentially discriminatory rules, that is, rules whose premises include the features in A. As shown in Figure 3, the percentage of actually discriminatory rules depends on the threshold α we take for *elift*, and ranges from 21.2% (for $\alpha = 1$) to 2.5% (for $\alpha = 1.5$).

5.2 Results of the method with no simulator

We also conducted an experiment to assess the effect of RR on the accuracy of the rules mined from the randomized predictions.

We want to test if, after using RR to protect some values from these responses, the resulting trained explainable model is close to a model trained on the raw, unaltered responses.

For that purpose we performed RR on 80% of the Adult data set for the Cartesian product of attributes *Sex*, *Education* and *Race*, so that the probability matrix had one row and one column for each combination of the three attributes.

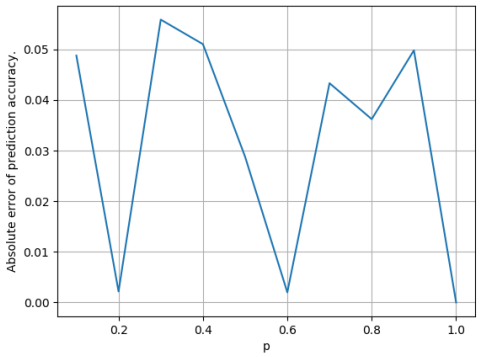


Figure 4: Prediction error (%) with rules mined from data sets randomized with probability p of leaving values unaltered

Specifically, we obtained 10 randomized versions of Adult, each with a different probability matrix $\mathbf{P}_p$, for $p = 0.1, 0.2, 0.3, \cdots, 1.0$. Matrix $\mathbf{P}_p$ had all the probabilities in its diagonal set to p and the off-diagonal probabilities set as described in Section 6.2 of [3], that is,

$$p_{uv} = (1-p)\frac{d_{uv}}{\sum_k d_{uk}},$$

where d_{uv} is the inverse of the number of differing attribute values between u and v, and $\sum_k d_{uk}$ is the sum of all changes between feature set u and all other feature sets k.

After that, we computed the estimated distributions of the data set as per Equation (1) and reconstructed the data sets by sampling random feature sets from the obtained distributions. Note that the estimated distributions can be computed from the randomized data sets and the matrices $\mathbf{P}_p$. From each reconstructed data set, we followed the substeps of Step 2 of Protocol 2, that yielded a set of mined rules. So we got 10 sets of mined rules for $p = 0.1, \cdots, 1.0$. Note that for $p = 1.0$ there was no real randomization and what was obtained were the rules mined on the original Adult data set, which we used as a baseline.

Then we evaluated the 9 sets of rules corresponding to $p = 0.1, \cdots, 0.9$ by computing their predictions (>50K, $\leq$50K) on the features of the remaining 20% of the Adult dataset and comparing them against the predictions given by the rules for $p = 1.0$. Figure 4 shows the percentage of Adult samples in which the 9 sets of rules gave predictions different from those of the baseline set of rules. Although the error may seem to vary a lot with p, in reality it varies within a very narrow range (0% to 5.5%).

6 CONCLUSIONS AND FUTURE WORK

We have presented a methodology that enables individuals to audit deep models that make decisions on them. The novelty of our approach is that it works even if the service provider owning the deep model restricts access to it in order to preserve its trade secret. To circumvent access restrictions, we adopt a collaborative approach. We consider two scenarios, with and without a model simulator.

When there is no simulator, we propose to use randomized response to allow users to share information without renouncing their privacy.

Future work will include testing on more data sets, fine tuning the parameters of rule elicitation and randomized response, and increasing the number of utility metrics tried. We will also deal with the dimensionality problem that arises in randomized response of Cartesian products when there are many features involved in decisions. Finally, implementing the proposed collaborative protocols with a user-friendly interface would go a long way towards effectively empowering the general public with audit capabilities.

ACKNOWLEDGMENTS AND DISCLAIMER

The following funding sources are gratefully acknowledged: European Commission (project H2020 700540 "CANVAS"), Government of Catalonia (ICREA Acadèmia Prize to J. Domingo-Ferrer and grant no. 2017 SGR 705), Spanish Government (project TIN2014-57364-C2-R "SmartGlacis"). The authors are with the UNESCO Chair in Data Privacy, but the views in this paper are their own and do not necessarily reflect those of UNESCO.

REFERENCES

[1] A. Chaudhuri and R. Mukerjee. *Randomized Response: Theory and Techniques*. Marcel Dekker, 1988.

[2] J. Domingo-Ferrer, S. Martínez, D. Sánchez and J. Soria-Comas. Co-utility: self-enforcing protocols for the mutual benefit of participants. *Engineering Applications of Artificial Intelligence*, 59:148-158, 2017.

[3] J. Domingo-Ferrer, R. Mulero-Vellido, and J. Soria-Comas. Multiparty computation with statistical input confidentiality via randomized response. In *Privacy in Statistical Databases-PSD 2018*, pp. 175–186. Springer, 2018.

[4] *General Data Protection Regulation*. Regulation (EU) 2016/679. https://gdpr-info.eu

[5] S. Greengard. Weighing the impact of GDPR. *Communications of the ACM*, 61(11):16–18, 2018.

[6] S. Hajian and J. Domingo-Ferrer. A methodology for direct and indirect discrimination prevention in data mining. *IEEE Transactions on Knowledge and Data Engineering*, 25(7):1445–1459, 2013.

[7] S. Hajian, J. Domingo-Ferrer, and O. Farràs. Generalization-based privacy preservation and discrimination prevention in data publishing and mining. *Data Mining and Knowledge Discovery*, 28(5-6):1158–1188, 2014.

[8] S. Hajian, J. Domingo-Ferrer, A. Monreale, D. Pedreschi, and F. Giannotti. Discrimination-and privacy-aware patterns. *Data Mining and Knowledge Discovery*, 29(6):1733–1782, 2015.

[9] The European Comission's High-Level Expert Group on Artificial Intelligence. *Draft Ethics Guidelines for Trustworthy AI*. December 2018.

[10] D. H. Park, L. A. Hendricks, Z. Akata, B. Schiele, T. Darrell, and M. Rohrbach. Attentive explanations: Justifying decisions and pointing to the evidence. *arXiv preprint arXiv:1612.04757*, 2016.

[11] D. Pedreschi, S. Ruggieri, and F. Turini. Measuring discrimination in socially-sensitive decision records. In *Proceedings of the 2009 SIAM International Conference on Data Mining*, pp. 581–592. SIAM, 2009.

[12] D. Pedreschi, S. Ruggieri, and F. Turini. Discrimination-aware data mining. In *14th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, pp. 560–568. ACM, 2008.

[13] B. Poulin, R. Eisner, D. Szafron, P. Lu, R. Greiner, D. S. Wishart, A. Fyshe, B. Pearcy, C. MacDonell, and J. Anvik. Visual explanation of evidence with additive classifiers. In *21st AAAI Conf. on Artificial Intelligence–AAAI'06*, pp. 1822-1829. AAAI, 2006.

[14] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should I trust you?: Explaining the predictions of any classifier. In *22nd ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, pp. 1135–1144. ACM, 2016.

[15] M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In *32nd AAAI Conf. on Artificial Intelligence–AAAI'18* pp. 1527-1535. AAAI, 2018.

[16] A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. *arXiv preprint arXiv:1704.02685*, 2017.

[17] S. Singh, M. T. Ribeiro, and C. Guestrin. Programs as black-box explanations. *arXiv preprint arXiv:1611.07579*, 2016.

[18] E. Strumbelj and I. Kononenko. An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, 11:1–18, 2010.

[19] R. Turner. A model explanation system. In *IEEE Intl. Workshop on Machine Learning for Signal Processing–MLSP'16*. IEEE, 2016.

[20] M. M. C. Vidovic, N. Görnitz, K.-R. Müller, and M. Kloft. Feature importance measure for non-linear learning algorithms. *arXiv preprint arXiv:1611.07567*, 2016.

[21] S. Wachter, B. Mittelstadt, and L. Floridi. Why a right to explanation of automated decision-making does not exist in the general data protection regulation. *International Data Privacy Law*, 7(2):76–99, 2017.

[22] S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

[23] M. B. Zafar, I. Valera, M. Gomez Rodriguez, and K. P. Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proc. of 26th Intl. Conf. on World Wide Web*, pp. 1171–1180. 2017.