Classification with Strategically Withheld Data

Anilesh Krishnaswamy [1] **Haoming Li** [1] **David Rein** [1] **Hanrui Zhang** [1] **Vincent Conitzer** [1]

Abstract

Machine learning techniques can be useful in applications such as credit approval and college admission. However, to be classified more favorably in these contexts, an agent may decide to strategically withhold some of her features, such as bad test scores. This is a missing data problem with a twist: which data is missing *depends on the chosen classifier*, because the specific classifier is what may create the incentive to withhold certain feature values. In this paper, we address the problem of training classifiers that are robust to this behavior.

We design two classification methods: MINCUT and HILL-CLIMBING (HC). We show that MINCUT is optimal when the true distribution of data is fully known. However, it can produce complex decision boundaries, and hence be prone to overfitting in some cases. Based on a characterization of truthful classifiers that are truthful (i.e., give no incentive to strategically hide features), we devise a simpler alternative called HC which consists of a hierarchical ensemble of out-of-the-box classifiers, trained using a specialized hill-climbing procedure which we show to be convergent. We also show that our algorithms perform well in experiments on real-world data sets, especially when features are suitably discretized.

1. Introduction

Applicants to most colleges in the US are required to submit their scores for at least one of the SAT and the ACT. Both tests are more or less equally popular, with close to two million taking each in 2018.[1] Applicants usually take one of these two tests – whichever they think is more suited to their own tastes.[2] However, given the growing competitiveness of college admissions, many applicants now take both tests and then strategically decide whether to drop one of the scores (if they think it will hurt their application) or report both.[3] The key issue here is that it is impossible to distinguish between an applicant who takes both tests but reports only one, and an applicant that takes only one test—for example because the applicant simply took the one required by her school, the dates for the other test did not work with her schedule, or for other reasons that are not strategic in nature.[4]

Say a college wants to take a principled machine learning approach to making admission decisions based on the scores from these two tests. For simplicity, assume no other information is available. Assume that the college has enough historical examples that contain the scores of individuals (on whichever tests are taken, truthfully reported) along with the corresponding ideal (binary) admission decisions.[5] Based on this data, the college has to choose a decision function that determines which future applicants are accepted. If this function is known to the applicants, they are bound to strategize and use their knowledge of the decision function to decide the scores they report.[5] How can the classifier be trained to handle strategic reporting of scores at prediction time?

To see the intricacies of this problem, let us consider a simple example.

Example 1. *Say the scores for each of the two tests (SAT and ACT) take one of two values: H (for High) or L (for Low). Let $*$ denote a missing value. Then there are eight possible inputs (excluding $(*,*)$ since at least one score is required): (H,H), (H,L), (L,H), (L,L), $(H,*)$, $(*,H)$, $(L,*)$ and $(*,L)$. Assume the natural distribution (without any withholding) over these inputs is known, and so are the conditional probabilities of the label $Y \in \{0,1\}$, as shown below:*

Assume $Y = 1$ is the more desirable "accept" decision. Then, ideally, we would like to predict $\widehat{Y} = 1$ whenever $X \in$

[1]Department of Computer Science, Duke University, Durham, USA. Correspondence to: Anilesh Krishnaswamy <anilesh@cs.duke.edu>, Haoming Li <haoming.li@duke.edu>, David Rein <irving.rein@duke.edu>, Hanrui Zhang <hrzhang@cs.duke.edu>, Vincent Conitzer <conitzer@cs.duke.edu>.

[1] https://www.edweek.org/ew/articles/2017/05/24/in-race-for-test-takers-act-outscores-sat--for.html

[2] https://www.princetonreview.com/college/sat-act

[3] https://blog.collegevine.com/should-you-submit-your-sat-act-scores/

[4] https://blog.prepscholar.com/do-you-need-to-take-both-the-act-and-sat

[5] We make these assumptions more generally throughout the paper.

Table 1. True distribution of inputs and targets: $*$ denotes a missing value.

X	(H,H)	(H,L)	(L,H)	(L,L)	$(H,*)$	$(*,H)$	$(L,*)$	$(*,L)$
$Pr(X)$	$1/8$	$1/8$	$1/8$	$1/8$	$1/8$	$1/8$	$1/8$	$1/8$
$Pr(Y=1 \mid X)$	0.9	0.7	0.3	0.1	0.6	0.6	0.2	0.2
$Pr(Y=0 \mid X)$	0.1	0.3	0.7	0.9	0.4	0.4	0.8	0.8

$\{(H,H),(H,L),(H,*),(*,H)\}$. *However, the strategic reporting of scores at prediction time effectively means, for example, that an input* $(*,H)$ *cannot be assigned the accept decision of* $\widehat{Y}=1$ *unless the same is done for* (L,H) *as well; otherwise, someone with* (L,H) *would simply not report the first test, thereby misreporting* $(*,H)$ *and being accepted. Taking this into account, the optimal classifier is given by* $\widehat{Y}=1$ *whenever* $X \in \{(H,H),(H,L),(H,*)\}$

There are many other settings where a similar problem arises. Many law schools now allow applicants to choose between the GRE and the traditional LSAT.[6] Recently, as a result of the COVID-19 pandemic, universities have implemented optional pass/fail policies, where students can choose to take some or all of their courses for pass/fail credit, as opposed to a standard letter grade that influences their GPA. They are often able to decide the status after already knowing their performance in the course. For credit scoring, some individuals or enterprises might not report some of their information, especially if this is not mandatory by law (Florez-Lopez, 2010).

The ability of strategic agents to withhold some of their features at prediction time poses a challenge only when the data used to train the classifier has some naturally missing components to begin with. For if not, the *principal* – e.g., the entity deciding on admissions – can decide to reject all agents that withhold any of their features, thereby forcing them to reveal all features. We focus on how a principal can best train classifiers that are robust even when there is strategic withholding of data by agents. Our methods will produce classifiers that eliminate the incentive for agents to withhold data.

Our contributions We now describe the key questions we are faced with, and how we answer them. Our model is described formally in *Section 2*. All our proofs are in the Supplement.

If the true input distribution is known, can we compute the optimal classifier? (Section 3) We answer this question in the affirmative by showing that the problem of computing the optimal classifier (Theorem 1) in this setting reduces to the classical Min-cut problem (Cormen et al., 2009). This analysis also gives us the MINCUT classifier, which can be computed on the empirical distribution, estimated using whatever data is available. However, since it can potentially give complex decision boundaries, it might not generalize well.

Are there simpler classifiers that are robust to strategic withholding of features? (Section 4) We first characterize the structure of classifiers that are "truthful", i.e., give no incentive to strategically hide features at prediction time (Theorem 2). Using this characterization, we devise a hill-climbing procedure (HC) to train a hierarchical ensemble of out-of-the-box classifiers and show that the procedure converges (Theorem 4) as long as we have black-box access to an agnostic learning oracle. We also analytically bound the generalization error of HC (Theorem 3). The ensemble of HC can be populated with any commonly used classifiers such as logistic regression, ANNs, etc.

How do our methods perform on real data sets? (Section 5) We conduct experiments on several real-world data sets to test the performance of our methods, comparing them to each other, as well as to other methods that handle missing data but ignore the strategic aspect of the problem. Our methods perform well, especially when we suitably discretize the data, thereby reducing overfitting.

Related work Our work falls broadly in the area of *strategic machine learning*, wherein a common assumption is that strategic agents can modify their features (i.e., misreport) in certain ways (normally at some cost), either to improve outcomes based on the classifier chosen by the principal (Hardt et al., 2016) or to influence which classifier is chosen in the first place (Dekel et al., 2010a). The main challenge in strategic machine learning, as in this paper, is the potential misalignment between the interests of the agents and the principal. Existing results in this line of work (Chen et al., 2018; Kleinberg & Raghavan, 2019; Haghtalab et al., 2020), often mainly theoretical, consider classifiers of a specific form, say linear, and ways of misreporting or modifying features in that context. Our results are different in that we focus on a specific type of strategic misreporting, i.e., withholding parts of the data, and devise general methods that are robust to this behavior that, in addition to having theoretical guarantees, can be tested practically, as done in this paper.

Our problem can also be viewed as an instance of *automated mechanism design with partial verification* (Green & Laffont, 1986; Yu, 2011; Kephart & Conitzer, 2015; 2016) where it is typically assumed that the feature space (usually called type space in mechanism design) is discrete and

[6] https://www.ets.org/gre/revised_general/about/law/

has reasonably small cardinality, and a prior distribution is known over the feature space. In contrast, the feature spaces considered in this paper consist of all possible combinations of potentially continuous feature values. Moreover, the population distribution can only be accessed by observing examples. Thus, common methodologies in automated mechanism design do not suffice for our setting.

A set of closely related (in particular, to Theorem 1) theoretical results are those of Zhang et al. [2019b; 2019a] on the problem of distinguishing "good" agents from "bad" (where each produces a different distribution over a sample space, and the agent can misreport the set of n samples that she has drawn). However, our work is different in that we consider the standard classification problem, we focus more on practical aspects, and we do not rely on the full knowledge of the input distribution.

Our work also finds a happy intersection between strategic machine learning and the literature on classification with missing data (Batista & Monard, 2003; Marlin, 2008). We devise methods that can deal with the strategic withholding by agents of some of their features, against a backdrop of missing data caused by natural reasons (e.g., nonstationary distributions of which features are present, or input sensor failures). Our hill-climbing algorithm can be viewed as an ensemble method for missing data (Conroy et al., 2016) that is strategy-proof against the aforementioned strategic behavior. We also study the performance of other standard, non-strategic classification methods for missing data in the strategic setting, including predictive value imputation and reduced-feature modeling (Saar-Tsechansky & Provost, 2007).

The problem we study is also connected to *adversarial classification* (Dalvi et al., 2004; Globerson & Roweis, 2006; Dekel et al., 2010b). We discuss this in more detail in the Supplement.

2. Preliminaries

We now describe our model and define notation required for our results in the ensuing sections.

Model with strategically withheld features: We have an input space $\mathcal{X}$, a label space $\mathcal{Y} = \{0, 1\}$, and a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ which models the population. A classifier $f : \mathcal{X} \to \mathcal{Y}$ maps a combination of features to a label. Let $F = [k] = \{1, \ldots, k\}$ be the set of features, each of which a data point may or may not have. For a data point $x \in \mathcal{X}$, let x_i denote the value of its i-th feature ($x_i = *$ if x does not have feature $i \in [k]$). For any set of features $S \subseteq [k]$, define $x|_S$ to be the projection of x onto S (i.e., retain features in S and drop those not in S):

$$(x|_S)_i = \begin{cases} x_i, & \text{if } i \in S \\ *, & \text{otherwise.} \end{cases}$$

We assume that data can be strategically manipulated at prediction (test) time in the following way: an agent whose true data point is x can report any other data point x' such that $x|_S = x'$ for some $S \subseteq [k]$. We use $\to$ to denote the relation between any such pair x, x' ($x \to x' \iff \exists S \subseteq [k] : x|_S = x'$). Note that $\to$ is transitive, i.e., for any $x_1, x_2, x_3 \in \mathcal{X}$, $x_1 \to x_2$ and $x_2 \to x_3 \implies x_1 \to x_3$.

We assume agents prefer label 1 to 0: in response to a classifier f, an agent with data point x will always withhold features to receive label 1 if possible, i.e., the agent will report $x' \in \operatorname{argmax}_{x'' : x \to x''} f(x'')$. Incorporating such strategic behavior into the loss of a classifier f, we get

$$\ell_{\mathcal{D}}(f) = \Pr_{(x,y)\sim\mathcal{D}} \left[y \neq \max_{x' : x \to x'} f(x') \right].$$

Truthful classifiers We will also be interested in *truthful* classifiers, which provably eliminate incentives for such strategic manipulation. A classifier f is *truthful* if for any $x, x' \in \mathcal{X}$ where $x \to x'$, $f(x) \geq f(x')$. In other words, not withholding any features is always an optimal way to respond to a truthful classifier. As a result, the loss of any truthful classifier f in the presence of strategically withheld features has the standard form: $\ell_{\mathcal{D}}(f) = \Pr_{(x,y)\sim\mathcal{D}}[f(x) \neq y]$.

Note that the so-called Revelation Principle – which states that in the presence of strategic behavior, any classifier f is equivalent to a truthful classifier f' – holds in this case because the reporting structure is transitive (see (Zhang et al., 2019a), details in appendices). In other words, we are guaranteed that, for any classifier f, there exists a truthful classifier f', such that for any $x \in \mathcal{X}$, $\max_{x' : x \to x'} f(x') = f'(x)$. Therefore, we focus on truthful classifiers in our model, without loss of generality.

3. Known Distributions and the MINCUT Classifier

We first present a method for computing an optimal classifier *when the input distribution is fully known*. Assuming $\mathcal{X}$ is finite, our goal is to characterize a classifier f^* which minimizes the loss $\ell_{\mathcal{D}}(.)$, for a known input distribution $\mathcal{D}$. As shorthand, define, for all $x \in \mathcal{X}$,

Definition 1. $\mathcal{D}^+(x) \coloneqq \Pr_{(x',y')\sim\mathcal{D}}[x' = x \wedge y' = 1]$, $\mathcal{D}^-(x) \coloneqq \Pr_{(x',y')\sim\mathcal{D}}[x' = x \wedge y' = 0]$.

The basic idea here is simple: we need to partition $\mathcal{X}$ into two sides, one labeled 1 and the other 0, where the error ac-

crued for each $x \in \mathcal{X}$ is given by $\mathcal{D}^+(x)$ or $\mathcal{D}^-(x)$, according to whether x is labeled 1 or 0. Such a partition should crucially respect the constraints imposed by the strategic behavior of agents, i.e., if $x \to x'$, then either x is labeled 1 or x' is labeled 0.

Definition 2. Given $\mathcal{X}$ and $\mathcal{D}$, let $G(\mathcal{D}, \mathcal{X})$ be a directed capacitated graph with vertices $V = \mathcal{X} \cup \{s, t\}$, where the edges E and edge capacities u are defined as follows:

- For each $x \in \mathcal{X}$, there is an edge $(s, x) \in E$ with capacity $u(s, x) = \mathcal{D}^-(x)$, and an edge $(x, t) \in E$ with capacity $u(x, t) = \mathcal{D}^+(x)$.
- For all pairs $x, x' \in \mathcal{X}$ such that $x \to x'$, there is an edge $(x', x) \in E$ with capacity $u(x', x) = \infty$.

In terms of the graph defined above, computing the optimal classifier f^* we seek is equivalent to finding a minimum s-t cut on $G(\mathcal{D}, \mathcal{X})$. The intuition is that the edges from s and to t reflect the value gained from labeling an example 0 or 1, respectively; one of the edges must be cut, reflecting the loss of not assigning it to the corresponding side. Moreover, if $x \to x'$, then the corresponding edge with infinite capacity prevents assigning x to the 0 side and x' to the 1 side.

Theorem 1. *If $(S, \bar{S})$ is a minimum s-t cut of $G(\mathcal{D}, \mathcal{X})$ (where S is on the same side as s), then for the classifier $f^*(x) \coloneqq \mathbb{1}(x \in \bar{S})$, we have $\ell_{\mathcal{D}}(f^*) = \min_f \ell_{\mathcal{D}}(f)$.*

We note that, consequently, the optimal classifier can be computed in $\mathrm{poly}(|\mathcal{X}|)$ time. In practice, it is natural to expect that we do not know $\mathcal{D}$ exactly, but have a finite number of samples from it. A more practical option is to apply Theorem 1 to the empirical distribution induced by the samples observed, and hope for the classifier computed from that to generalize to the true population distribution $\mathcal{D}$.

The MINCUT Classifier Given a set $\widehat{\mathcal{X}}$ of m i.i.d. samples from $\mathcal{D}$, let $\bar{\mathcal{X}} \coloneqq \widehat{\mathcal{X}} \cup \{x' : x' \to x, \exists x \in \widehat{\mathcal{X}}\}$ and $\widehat{\mathcal{D}}$ be the corresponding empirical distribution over $\bar{\mathcal{X}}$. The MINCUT classifier is then obtained by applying Theorem 1 to $G(\widehat{\mathcal{D}}, \bar{\mathcal{X}})$. At test time, samples not in $\bar{\mathcal{X}}$ are given the 0 label. It can be seen that MINCUT is a truthful classifier. In light of traditional wisdom, the smaller m is relative to $\mathcal{X}$, the larger the generalization error of MINCUT will be. We do not attempt a theoretical analysis in this regard, but note that when $\mathcal{X}$ is large, the generalization error can be extremely large (see Example 2 in the Supplement). Therefore, a suitable discretization of features is sometimes useful, as we shall see in Section 5.

4. Truthful classifiers and HILL-CLIMBING

The other drawback of MINCUT, related to the issue of generalization just discussed, is that the decision boundary it generates is potentially complex, and therefore hard to interpret meaningfully in a practical setting. In this section, we devise a simpler alternative called HILL-CLIMBING. To help introduce this algorithm, we first present a characterization of truthful classifiers in our setting, since we can limit our focus to them without loss of generality (as discussed in Section 2). For shorthand, we use the following definition:

Definition 3 (F'-classifier)**.** For a subset of features $F' \subseteq F$, a classifier f is said to be an F'-classifier if for all $x \in \mathcal{X}$, we have $f(x) = f(x|_{F'})$ and $x_i = * \implies f(x) = 0$.

In other words, an F'-classifier depends only on the values of the features in F', rejecting any x where any of these is empty. We can collect many such classifiers into an ensemble as follows:

Definition 4 (MAX Ensemble)**.** For a collection of classifiers $\mathcal{C} = \{f_j\}$, its MAX Ensemble classifier is given by $\textsc{Max}_{\mathcal{C}}(.) \coloneqq \max_j f_j(.)$.

This is equivalent to getting each agent to pick the most favorable classifier from among those in $\{f_j\}$. Now using the above definitions we have the following characterization of truthful classifiers:

Theorem 2. *A classifier f is truthful iff $f(.) = \textsc{Max}_{\mathcal{C}}(.)$ for a collection of classifiers $\mathcal{C} = \{f_j\}$ such that, for some $\{F_j\} \subseteq 2^F$, each f_j is an F_j-classifier .*

Now, for any truthful classifier f, we can bound the gap between its population loss $\ell_{\mathcal{D}}(f)$ and its empirical loss on a set of samples $\widehat{\mathcal{X}}$ denoted by $\ell_{\widehat{\mathcal{X}}}(f) \coloneqq \frac{1}{m} \sum_{i \in [m]} |f(x_i) - y_i|$. Before stating a theorem to this end, we define the following entities: Let $\mathcal{H}$ be a base hypothesis space over $\mathcal{X}$, and $n \in \{1, \ldots, 2^k\}$ be a parameter. Define $d \coloneqq d_{\mathrm{VC}}(\mathcal{H})$ to be the VC dimension of $\mathcal{H}$. Define $\bar{\mathcal{H}}$ as the set of all classifiers that can be written as the MAX Ensemble of n classifiers in $\mathcal{H}$.

Theorem 3. *Let $\widehat{\mathcal{X}} = \{(x_i, y_i)\}_{i \in [m]}$ be m i.i.d. samples from $\mathcal{D}$. For any $f \in \bar{\mathcal{H}}$, for any $\delta > 0$, with probability at least $1 - \delta$, we have $\ell_{\mathcal{D}}(f) \leq \ell_{\widehat{\mathcal{X}}}(f) + O\left(\sqrt{\frac{dn \cdot \log dn \cdot \log m + \log(1/\delta)}{m}}\right)$.*

It is easy to see that for any of the common hypothesis spaces used in practice – say $\mathcal{H}$ consists of linear hypotheses – if a truthful classifier f is in $\mathcal{H}$, then so are the components of the MAX Ensemble version of f as in Theorem 2. We have, however, stated Theorem 3 in slightly more general terms.

The HILL-CLIMBING classifier We now present a hill-climbing approach with provable convergence and generalization guarantees. The HILL-CLIMBING classifier (henceforth HC) builds upon the characterization of truthful classifiers in Theorem 2. Intuitively, the approach works by

considering a hierarchy of classifiers, organized by the features involved. For example, consider a setting with $k = 3$ features. We make a choice as to what classifiers we use — say f_1 for input of the form $(x_1, *, *)$, f_2 for input of the form $(x_1, x_2, *)$, and f_3 for input of the form (x_1, x_2, x_3). Any agent with features 1 and 2 (but not 3), for example, should be able to report both features so as to be classified by f_2, or feature 2 to be classified by f_1 instead. So in effect, assuming full knowledge of the classifiers, each agent can check all of the classifiers and choose the most favorable one. Without loss of generality, we assume that when a data point does not have all the features required by a classifier, the outcome is automatically a rejection.

In short, HC (defined formally in Algorithm 1) works as follows: first choose a hypothesis space $\mathcal{H}$, in order for Theorem 3 to apply. Then select n subsets of F (where n is a parameter), say $F_1, F_2, \ldots, F_n$. For each F_j, we learn a F_j-classifier, say f_j, from among those in H. Start by initializing these classifiers to any suitable $\{f_1^0, \ldots, f_n^0\}$. In each iterative step, each of the basic classifiers is updated to minimize the empirical loss on the samples that are rejected by all other classifiers. We next show that such an update procedure always converges. To do so, as far as our theoretical analysis goes, we assume we have black-box access to an agnostic learning oracle (Line 1 in Algorithm 1). After convergence, the HC classifier is obtained as the MAX Ensemble of these classifiers. The generalization guarantee of Theorem 3 applies directly to the HC classifier.

Theorem 4. *Algorithm 1 converges.*

Algorithm 1 HILL-CLIMBING (HC) Classifier

Input: data set $\widehat{\mathcal{X}} = \{(x_i, y_i)\}_{i\in[m]}$, n subsets $F_1, F_2, \ldots, F_n$ of F.
Initialize: $t \leftarrow 0$, $\{f_1^0, \ldots, f_n^0\}$.
while $\Delta > 0$ **do**
  for $i = 1, 2, \ldots, n$ **do**
    $S_i \leftarrow \{(x, y) \in \widehat{\mathcal{X}} : f_j^t(x|_{F_j}) = 0, \forall j \neq i\}$.
    $f_i^{t+1} = \text{argmin}_{f\in\mathcal{H}} \sum_{(x,y)\in S_i} |f(x|_{F_i}) - y|$.
  end for
  $f^* \leftarrow \text{MAX}_{\{f_1^{t+1},\ldots,f_n^{t+1}\}}$; $\ell_t = \ell_{\widehat{\mathcal{X}}}(f^*)$
  $\Delta \leftarrow \ell_t - \ell_{t-1}$; $t \leftarrow t + 1$
end while
Return: f^*.

Implementing HC**.** In practice, the basic classifiers $\{f_1, f_2, \ldots, f_n\}$ in HC can be populated with any standard out-of-the-box classifiers such as logistic regression classifiers or neural networks, the choice of which can influence the performance of f. In Section 6, we test HC with a few such options. The assumption of having access to an agnostic learning oracle does not play a crucial role in practice, with standard training methods performing well enough to ensure convergence. We also note that we are free to choose any $F_1, F_2, \ldots, F_n$ to define HC – we perform feature selection on datasets with a large number of features, and use all possible subsets of the selected features (see Section 5).

5. Evaluation

In this section, we compare the performance of MINCUT and a few variants of HC against several out-of-the-box counterparts (that do not account for the strategic withholding of data). We show that our methods provide a significant advantage when such strategic behavior is at play.

Datasets Four credit approval datasets are obtained from the UCI repository (Dua & Graff, 2017) – one each from Australia, Germany, Poland and Taiwan. Due to computational constraints in simulating the combinatorial strategic behavior, we filter each dataset to only the top 4 features according to ANOVA F-value. The variation, across the datasets, of the types of features thus obtained, provides us with a wide variety of testing conditions. Also, as is commonplace, these datasets are imbalanced to various degrees. In order to demonstrate the performance of classifiers in a standard, controlled setting, we balance them via random undersampling. The basic characteristics of the datasets before and after these two steps of preprocessing are summarized in Table 2.

We then randomly remove a fraction $\epsilon = 0, 0.1, \ldots, 0.5$ of all feature values in each dataset to simulate data that is missing "naturally" – i.e., not due to strategic withholding.

Classifiers We evaluate MINCUT, and variants of HC that use logistic regression (LR) and neural networks (ANN). We select LR for its popularity in credit scoring and ANN for being the best-performing individual classifiers on credit approval datasets (Lessmann et al., 2015).

For the purposes of comparison, we include MAJ – predict the *majority* label if examples with the exact same feature values appeared in the training set, and reject if not – which can be thought of as a non-strategic counterpart of MINCUT. We also include standard methods like kNN (k-Nearest Neighbors, which is somewhat similar to MAJ), and others such as ANN and LR. Since HC is an ensemble method, RF (Random Forest), the best-performing homogeneous ensemble on credit approval datasets (Lessmann et al., 2015), is added to the mix. We use these out-of-the-box methods (kNN, LR, ANN, RF) by employing either mean/mode imputation (Lessmann et al., 2015) or reduced-feature modeling (Saar-Tsechansky & Provost, 2007), to deal with missing data. We use MINCUT, HC, and MAJ with and without discretization (Fayyad & Irani, 1992). For the sake of our exposition here, we do not use discretization

Table 2. Dataset summary statistics

Dataset	Size	Total # of features	Size after balancing	Features after selection
Australia	690	15	614	2 numerical, 2 categorical
Germany	1000	20	600	1 numerical, 3 categorical
Poland (5-year)	5910	64	820	4 numerical
Taiwan	30,000	23	13,272	4 ordinal

for Imputation and Reduced Feature methods, as it does not really help (see **Results** below, and also the Supplement).

Testing We test all methods under two ways of reporting: "truthful", i.e., all features are reported as is, and "strategic", i.e., some features might be withheld if it leads to a better outcome. We measure the test accuracy of each classifier, averaged over N=100 runs, with randomness over the undersampling and the data that is randomly chosen to be missing.[7] Other metrics, and details about implementing and training the classifiers, are discussed in the Supplement.

Results In Table 3, we present the mean accuracy (across the randomization just discussed) of all the methods we consider, with both truthful and strategic reporting, for each of the four data sets (for one value of $\epsilon = 0.2$). The main takeaways from these numbers are the following: • At least one of the HC and MINCUT variants outperforms all out-of-the-box classifiers under strategic reporting.. • Discretization helps HC and MINCUT when the features are mostly continuous (e.g., Poland), and naturally, not when the data is already discrete (e.g., Taiwan). And as expected, under truthful reporting, the standard out-of-the-box methods perform better, especially LR (Imputation). However, LR (Imputation, and also, Reduced Feature) do not perform well under strategic reporting (especially in Germany and Taiwan, where the data is more discrete), as we discuss next.

In Figure 1, we present a box plot of the accuracy of various classifiers across our experiments, under strategic reporting. For the sake of exposition, out of all the out-of-box-methods, we include just LR (both Imputation and Reduced Feature variants). In addition to reinforcing the above-mentioned takeaways, Figure 1 leads to a few more interesting observations: • *Imputation-based methods* are sensitive in how they interact with the mean/mode value used in imputation: if the imputed values are generally a negative signal, then agents have little incentive to drop features, and these methods perform well (Fig. 1, Australia); if the imputed values are generally a positive signal, then agents would withhold features, and these methods perform worse (Fig. 1, Poland); and finally, if the imputed values do not give a clear signal (e.g., when the distribution of each feature value is not skewed), there is a high variance in the performance of these methods (Fig. 1, Germany). • *Reduced Feature* modeling, despite performing comparably to imputation under truthful reporting, allows too many examples to be accepted under strategic reporting, which significantly hurts its performance. This is true especially for smaller ϵ, since then each of the component classifiers have fewer examples to train on, and therefore, present several viable options for strategic withholding.

[7] to simulate data missing for non-strategic reasons.

A note on feature selection Given that we filter our data sets to just the top 4 features, we check the accuracy of the imputation classifiers under truthful reporting using all the features available (Table 4). We restrict ourselves to imputation methods here, since they are computationally tractable with a large number of features. When we do this for $\epsilon = 0, 0.2$, we find that the accuracy achieved is similar to when we use just the top 4 features (Tables 3, 4). Therefore, if we were somehow able to simulate strategic reporting on a large number of features (which is needed to test out-of-the-box methods), it is plausible that our methods working on just the 4 features will still do better.

6. Conclusion

In this paper, we studied the problem of classification when each agent at prediction time can strategically withhold some of its features to obtain a more favorable outcome. We devised classification methods that are robust to this behavior, i.e., eliminate the incentive for an agent to withhold data. We presented the MINCUT classifier, which is optimal given full knowledge of the input distribution, but can sometimes lead to overfitting. We also characterized the space of all possible truthful classifiers in our setting, showing that they can be thought of hierarchical ensembles of a specific kind. Building on this characterization, we defined a HC methodology for constructing such ensembles, which has provable generalization guarantees. We tested these methods on real-world data sets, showing that they outperform standard methods that do not account for the aforementioned strategic behavior.

In order to train a classifier using the HC methodology, we start by selecting a small number of features from the data. This comes at little cost on the data sets that we test on – most of the information useful for prediction is contained within these features. Therefore, most standard methods

Table 3. Our methods vs. the rest: mean classifier accuracy for $\epsilon = 0.2$ ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.793	.793	.640	.640	.659	.659	.648	.648
HC (LR) w/ disc.	.794	**.794**	.642	**.642**	.690	.690	.650	.650
HC (ANN)	.774	.774	.625	.625	.640	.640	.647	.647
HC (ANN) w/ disc.	.769	.769	.617	.617	.679	.679	.648	.648
MINCUT	.768	.768	.581	.581	.500	.500	.652	**.652**
MINCUT w/ disc.	.789	.789	.630	.630	.693	**.693**	.649	.649
MAJ	.665	.670	.539	.524	.500	.500	.678	.560
MAJ w/ disc.	.795	.618	.605	.533	.715	.565	**.688**	.566
LR (Imputation)	.796	.788	**.665**	.579	.715	.662	.670	.619
ANN (Imputation)	.798	.786	.665	.581	.704	.648	.686	.535
RF (Imputation)	.771	.630	.626	.561	**.722**	.603	.679	.549
kNN (Imputation)	.791	.722	.638	.566	.694	.626	.662	.560
LR (Reduced Feature)	**.809**	.544	.632	.508	.669	.510	.666	.592
ANN (Reduced Feature)	.806	.542	.629	.509	.661	.509	.668	.546
RF (Reduced Feature)	.779	.537	.611	.507	.698	.518	.679	.574
kNN (Reduced Feature)	.781	.531	.604	.505	.655	.510	.658	.586

Figure 1. Box plot of classifier accuracy: under strategic reporting

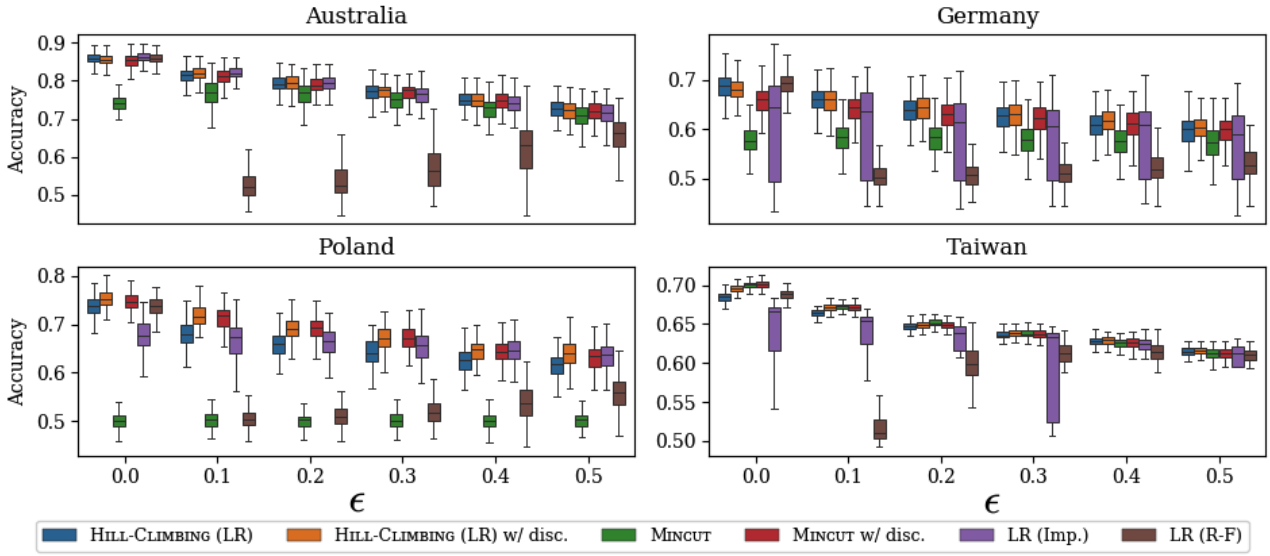

Table 4. Mean accuracy without feature selection: imputation methods under truthful reporting

Classifier	Australia		Germany		Poland		Taiwan	
	$\epsilon = 0$	$\epsilon = .2$	$\epsilon = 0$	$\epsilon = .2$	$\epsilon = 0$	$\epsilon = .2$	$\epsilon = 0$	$\epsilon = .2$
LR (Imp.)	.848	.803	.690	.660	.746	.733	.670	.657
ANN (Imp.)	.843	.802	.685	.654	.753	.740	.703	.690
RF (Imp.)	.869	.822	.702	.664	.812	.786	.704	.696
kNN (Imp.)	.811	.772	.655	.621	.743	.720	.667	.650

perform very similarly whether they are applied to all the features or just these. And it is natural to expect that using a smaller number of features diminishes the damage done by strategic behavior under most reasonable classifiers. The other reason is that having a small number of features makes it easy to train MINCUT and HC. Still, an interesting open question is to extend these techniques to settings in which it is important to use a large number of features.

There is an extremely rich literature on dealing with missing data in general (Rubin, 1976; Allison, 2001; Batista & Monard, 2003; Saar-Tsechansky & Provost, 2007). We complement this line of research by focusing on the specific case where data is missing for strategic reasons. An important question for future work is to develop a broader theory of robustness to missing data that naturally includes the case of strategic withholding.

Broader Impact

The methods presented in this paper are geared towards preventing the strategic withholding of data when machine learning methods are used in real-world applications. This will increase the robustness of ML techniques in these contexts: without taking this issue into account, deployment of these techniques will generally result in a rapid change in the distribution of submitted data due to the new incentives faced, causing techniques to work much more poorly than expected at training time. Thus, there is an AI safety (Amodei et al., 2016) benefit to our work. The lack of strategic withholding also enables the collection of (truthful) quality data. Of course, there can be a downside to this as well if the data is not used responsibly, which could be the case especially if the features that (without our techniques) would have been withheld are sensitive or private in nature.

References

Allison, P. D. *Missing data*, volume 136. Sage publications, 2001.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P. F., Schulman, J., and Mané, D. Concrete problems in ai safety. *ArXiv*, abs/1606.06565, 2016.

Batista, G. E. and Monard, M. C. An analysis of four missing data treatment methods for supervised learning. *Applied artificial intelligence*, 17(5-6):519–533, 2003.

Chawla, N., Japkowicz, N., and Kolcz, A. Icml2003 workshop on learning from imbalanced data sets (ii). In *Proceedings available at http://www. site. uottawa. ca/nat/Workshop2003/workshop2003. html*, 2003.

Chawla, N. V., Japkowicz, N., and Kotcz, A. Special issue on learning from imbalanced data sets. *ACM SIGKDD explorations newsletter*, 6(1):1–6, 2004.

Chen, Y., Podimata, C., Procaccia, A. D., and Shah, N. Strategyproof linear regression in high dimensions. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pp. 9–26, 2018.

Conroy, B., Eshelman, L., Potes, C., and Xu-Wilson, M. A dynamic ensemble approach to robust classification in the presence of missing data. *Machine Learning*, 102(3):443–463, 2016.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. *Introduction to algorithms*. MIT press, 2009.

Dalvi, N., Domingos, P., Sanghai, S., and Verma, D. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 99–108, 2004.

Dekel, O., Fischer, F., and Procaccia, A. D. Incentive compatible regression learning. *Journal of Computer and System Sciences*, 76(8):759–777, 2010a.

Dekel, O., Shamir, O., and Xiao, L. Learning to classify with missing and corrupted features. *Machine learning*, 81(2):149–178, 2010b.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Fayyad, U. M. and Irani, K. B. On the handling of continuous-valued attributes in decision tree generation. *Machine learning*, 8(1):87–102, 1992.

Florez-Lopez, R. Effects of missing data in credit risk scoring. a comparative analysis of methods to achieve robustness in the absence of sufficient data. *The Journal of the Operational Research Society*, 61(3):486–501, 2010.

Globerson, A. and Roweis, S. Nightmare at test time: robust learning by feature deletion. In *Proceedings of the 23rd international conference on Machine learning*, pp. 353–360, 2006.

Green, J. R. and Laffont, J.-J. Partially verifiable information and mechanism design. *The Review of Economic Studies*, 53(3): 447–456, 1986.

Haghtalab, N., Immorlica, N., Lucier, B., and Wang, J. Maximizing welfare with incentive-aware evaluation mechanisms. In *29th International Joint Conference on Artificial Intelligence*, 2020.

Hand, D. J. and Anagnostopoulos, C. When is the area under the receiver operating characteristic curve an appropriate measure of classifier performance? *Pattern Recognition Letters*, 34(5): 492–495, 2013.

Hardt, M., Megiddo, N., Papadimitriou, C., and Wootters, M. Strategic classification. In *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*, pp. 111–122, 2016.

Japkowicz, N. Aaai2000 workshop on learning from imbalanced data sets. *AAAI Tech Report, No. WS-00-05*, 2000.

Kephart, A. and Conitzer, V. Complexity of mechanism design with signaling costs. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pp. 357–365, 2015.

Kephart, A. and Conitzer, V. The revelation principle for mechanism design with reporting costs. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pp. 85–102, 2016.

Kleinberg, J. and Raghavan, M. How do classifiers induce agents to invest effort strategically? In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pp. 825–844, 2019.

Lessmann, S., Baesens, B., Seow, H.-V., and Thomas, L. C. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, 247(1):124–136, 2015.

Marlin, B. *Missing data problems in machine learning*. PhD thesis, Citeseer, 2008.

Rubin, D. B. Inference and missing data. *Biometrika*, 63(3): 581–592, 1976.

Saar-Tsechansky, M. and Provost, F. Handling missing values when applying classification models. *Journal of machine learning research*, 8(Jul):1623–1657, 2007.

Yu, L. Mechanism design with partial verification and revelation principle. *Autonomous Agents and Multi-Agent Systems*, 22(1): 217–223, 2011.

Zhang, H., Cheng, Y., and Conitzer, V. Distinguishing distributions when samples are strategically transformed. In *Advances in Neural Information Processing Systems*, pp. 3187–3195, 2019a.

Zhang, H., Cheng, Y., and Conitzer, V. When samples are strategically selected. In *International Conference on Machine Learning*, pp. 7345–7353, 2019b.

A. Further related work

Our work can, in a way, be thought of as studying an adversarial classification (see, e.g., Vorobeychik & Kantarcioglu, 2018) problem – in particular, a decision-time, white-box, targeted attack on binary classifiers, assuming that the only strategy available to the attacker is to remove feature values, and the attacker's goal is to maximize the number of instances classified as positive. In this regard, what we study is similar in spirit to some of the existing literature (Globerson & Roweis, 2006; Syed & Taskar, 2010; Dekel et al., 2010b) on adversarial classification.

For example, (Globerson & Roweis, 2006) consider a problem where, at test time, the attacker can set up to a certain number of features (say pixels in an image) to zero for each instance individually in a way that is most harmful to the classifier chosen. To be robust to such attacks, they devise convex programming based methods that avoid depending on small sets of features to learn the class structure. Our work is different in that we take a more game-theoretic approach to designing classifiers (including ensemble-based ones) that are fully resistant to the strategic withholding of features by agents (that prefer being labeled positively). Moreover, we make no assumptions on the actual structure of the feature space.

B. Proofs

Theorem 1. *If $(S, \bar{S})$ is a minimum s-t cut of $G(\mathcal{D}, \mathcal{X})$ (where S is on the same side as s), then for the classifier $f^*(x) \coloneqq \mathbb{1}(x \in \bar{S})$, we have $\ell_{\mathcal{D}}(f^*) = \min_f \ell_{\mathcal{D}}(f)$.*

Proof. First observe that any classifier f can be viewed equivalently as a subset of $\mathcal{X}$, given by

$$\{x \in \mathcal{X} \mid f(x) = 1\}.$$

Below, we use these interpretations, i.e., as a function or a subset, of a classifier interchangeably.

The loss of a truthful classifier f can then be written as

$$\begin{aligned}
\ell_{\mathcal{D}}(f) &= \Pr_{(x,y)\sim\mathcal{D}}[(x \in f \wedge y = 0) \vee (x \notin f \wedge y = 1)] \\
&= \Pr_{(x,y)\sim\mathcal{D}}[x \in f \wedge y = 0] + \Pr_{(x,y)\sim\mathcal{D}}[x \notin f \wedge y = 1] \\
&= \sum_{x\in f} \Pr_{(x',y)\sim\mathcal{D}}[x' = x \wedge y = 0] + \sum_{x\notin f} \Pr_{(x',y)\sim\mathcal{D}}[x' = x \wedge y = 1] \\
&= \sum_{x\in f} \mathcal{D}^-(x) + \sum_{x\notin f} \mathcal{D}^+(x),
\end{aligned}$$

where the last line follows from Definition 1.

Therefore, our goal is to solve the following optimization problem:

$$\begin{aligned}
\min_{f\subseteq\mathcal{X}} \quad & \sum_{x\in f} \mathcal{D}^-(x) + \sum_{x\notin f} \mathcal{D}^+(x) \\
\text{s.t.} \quad & x' \in f \implies x \in f \qquad\qquad \forall x, x' \in \mathcal{X} \text{ where } x \to x'.
\end{aligned}$$

Consider the following min-cut formulation (using Definition 2): Let $G(\mathcal{D}, \mathcal{X})$ be a directed capacitated graph, with vertices $V = \mathcal{X} \cup \{s, t\}$, with edges E and edge capacities u defined as follows:

- For each $x \in \mathcal{X}$, there is an edge $(s, x) \in E$ with capacity $\mathcal{D}^-(x)$, and an edge $(x, t) \in E$ with capacity $\mathcal{D}^+(x)$.
- For each pair $x, x' \in \mathcal{X}$ where $x \to x'$, there is an edge $(x', x) \in E$ with capacity ∞.

Observe that each finite-capacity s-t cut $(S, \bar{S})$ corresponds bijectively to a truthful classifier $f \coloneqq \mathbb{1}(x \in \bar{S} \setminus \{t\})$. Moreover, the capacity of the cut is given precisely by

$$\sum_{x\in\bar{S}\cap\mathcal{X}} \mathcal{D}^-(x) + \sum_{x\in S\cap\mathcal{X}} \mathcal{D}^+(x) = \sum_{x\in f} D^-(x) + \sum_{x\notin f} D^+(x) = \ell_{\mathcal{D}}(f).$$

Therefore, any s-t min-cut corresponds to an optimal classifier f^*, which can be computed "efficiently" (i.e., in time $\text{poly}(|\mathcal{X}|)$) using any efficient max-flow algorithm given complete knowledge of $\mathcal{D}$. □

Theorem 2. *A classifier f is truthful iff $f(.) = \text{MAX}_{\mathcal{C}}(.)$ for a collection of classifiers $\mathcal{C} = \{f_j\}$ such that, for some $\{F_j\} \subseteq 2^F$, each f_j is an F_j-classifier.*

Proof. Recall that the revelation principle holds in our setting (as mentioned in Section 2, also see Proposition 1). It therefore suffices to characterize all direct revelation classifiers. For any (not necessarily truthful) classifier f, consider its direct revelation implementation f', which maps feature values x to the most desirable label the data point can get by dropping features, i.e.,

$$f'(x) = \max_{x':x\to x'} f(x').$$

We argue below that f' has the desired form.

Observe that depending on which features a data point x has, f can be decomposed into 2^k subclassifiers, denoted $\{f_F\}_{F\subseteq[k]}$. The label of x is then determined in the following way: let F_x be the set of features possessed by x, i.e.,

$$F_x = \{i \in [k] \mid x_i \neq *\}.$$

Then

$$f(x) = f_{F_x}(x).$$

Moreover, observe that (1) f_F effectively depends only on $x|_F$ (i.e., $f_F(x) = f_F(x|_F)$), since f_F only acts on those data points where all features not in F are missing, and (2) without loss of generality, f_F rejects any data point with a missing feature $i \in F$, since f_F never acts on a data point where such a feature $i \in F$ is missing. Now consider how f' works on a data point x. For any $F \subseteq F_x$, by dropping all features not in F, x can report $x_|F$. Moreover, for any such $F \subseteq F_x$, $f(x|_F) = f_F(x|_F)$. f' outputs 1 for x, iff there exists $F \subseteq F_x$, such that $f_F(x|_F) = 1$. One can therefore write f' in the following way: for any $x \in \mathcal{X}$,

$$f'(x) = \max_{F\subseteq[k]} f_F(x),$$

as desired. □

Theorem 3. *Let $\widehat{\mathcal{X}} = \{(x_i, y_i)\}_{i\in[m]}$ be m i.i.d. samples from $\mathcal{D}$. For any $f \in \bar{\mathcal{H}}$, for any $\delta > 0$, with probability at least $1-\delta$, we have $\ell_{\mathcal{D}}(f) \leq \ell_{\widehat{\mathcal{X}}}(f) + O\left(\sqrt{\frac{dn\cdot\log dn\cdot\log m+\log(1/\delta)}{m}}\right)$.*

Proof. Recall that $\bar{\mathcal{H}}$ is defined as the set of all classifiers that can be written as the MAX Ensemble of n classifiers in $\mathcal{H}$. Given the classical VC inequality (e.g., Shalev-Shwartz & Ben-David, 2014, Theorem 6.11), we only need to bound the VC dimension of $\bar{\mathcal{H}}$, and show that

$$d_{\text{VC}}(\bar{\mathcal{H}}) = O(dn \cdot \log dn),$$

where d is the VC dimension of $\mathcal{H}$. To this end, observe that each $f \in \bar{\mathcal{H}}$ is essentially a decision tree with $n+1$ leaves, where each leaf is associated with a binary label, and each internal node corresponds to a classifier in $\mathcal{H}$. To be precise, f can be computed in the following way: for any $x \in \mathcal{X}$, if $f_1(x) = 1$, then $f(x) = 1$; otherwise, if $f_2(x) = 1$, then $f(x) = 1$, etc. It is known (see, e.g., Daniely et al., 2015, Section 5.2) that the class of all such decision trees with $n+1$ leaves, which is a superset of $\bar{\mathcal{H}}$, has VC dimension $O(dn \log dn)$. As a result, $d_{\text{VC}}(\mathcal{H}) = O(dn \log dn)$, and the theorem follows. □

Theorem 4. *Algorithm 1 converges.*

Proof. Given $f = \text{MAX}_{\{f_1^t, f_2^t, \ldots, f_n^t\}}$, consider a single update step for, say, f_1^t. As in Algorithm 1, define:

$$S_1 = \{(x, y) \in \widehat{\mathcal{X}} : f_j^t(x|_{F_j}) = 0, \forall j \neq 1\},$$
$$S_{-1} = S \setminus S_1.$$

Then we perform the update as follows:

$$f_1^{t+1} = \text{argmin}_{h\in\mathcal{H}} \sum_{(x,y)\in S_1} |h(x|_{F_1}) - y|.$$

Let $f' = \textsc{Max}_{\{f_1^{t+1}, f_2^t, \ldots, f_n^t\}}$. Now, the loss calculated for f' is

$$\begin{aligned}
\ell_{\widehat{\mathcal{X}}}(f') &= \frac{1}{m}(|S_1| \cdot \ell_{S_1}(f') + |S_{-1}| \cdot \ell_{S_{-1}}(f')) \\
&= \frac{1}{m}(|S_1| \cdot \ell_{S_1}(f_1^{t+1}) + |S_{-1}| \cdot \ell_{S_{-1}}(f')) \\
&= \frac{1}{m}(|S_1| \cdot \ell_{S_1}(f_1^{t+1}) + |S_{-1}| \cdot \ell_{S_{-1}}(f)) \\
&\leq \frac{1}{m}(|S_1| \cdot \ell_{S_1}(f_1^{t}) + |S_{-1}| \cdot \ell_{S_{-1}}(f)) \\
&= \ell_{\widehat{\mathcal{X}}}(f).
\end{aligned}$$

The inequality in the above sequence of steps follows from the fact that f_j^{t+1} accrues a lower loss on S_1 than f_j^t by definition, and that the classification outcomes for any $(x, y) \in S_{-1}$ is the same for f and f'.

If we treat $\ell_{\widehat{\mathcal{X}}}(f)$ as a potential function, we can see that it can only decrease with each step, and therefore, the algorithm has to converge at some point. □

C. Revelation Principle

There are many results in the literature on partial verification as to the validity of the revelation principle in various settings (Green & Laffont, 1986; Yu, 2011; Kephart & Conitzer, 2015; 2016). For our purposes, as mentioned in Section 2, when the reporting structure is given by a partial order, the revelation principle holds. Below we give a quick proof for why this is the case in our setting.

Proposition 1. *For any classifier $f : \mathcal{X} \to \{0, 1\}$, there is a truthful classifier f' such that after misreporting, f and f' output the same label for all $x \in \mathcal{X}$, i.e.,*

$$f'(x) = \max_{x' : x \to x'} f(x').$$

Proof. Below we explicitly construct f'. Let f' be such that for $x \in \mathcal{X}$,

$$f'(x) = \max_{x' : x \to x'} f(x).$$

Clearly f' and f output the same label after strategic manipulation. We only need to show f' is truthful, i.e., for any $x_1, x_2 \in \mathcal{X}$ where $x_1 \to x_2$, $f'(x_1) \geq f'(x_2)$. Let $X_1 = \{x' : x_1 \to x'\}$ and $X_2 = \{x' : x_2 \to x'\}$. Recall that $\to$ is transitive and $x_1 \to x_2$, so $X_1 \supseteq X_2$. Now we have

$$f'(x_1) = \max_{x \in X_1} f(x) \geq \max_{x \in X_2} f(x) = f'(x_2).$$

□

D. Other observations

D.1. Regarding MINCUT

Naturally, the test error of MINCUT depends on $\mathcal{X}$ and m. For example, If $\mathcal{X}$ is discrete and small, one would expect that MINCUT is almost optimal given enough samples. However, when $\mathcal{X}$ is large or even infinite, the generalization gap can be extremely large. To see why this is true, consider the following example:

Example 2. *Say we are given a feature space with two features, each of which can take any real value between 0 and 1. Let the marginal distribution of $\mathcal{D}$ on $\mathcal{X}$ be the uniform distribution over $\mathcal{X} = \{(x, y), (x, *), (*, y) \mid x, y \in [0, 1]\}$. When we see a new data point (x, y), unless we already have (x, y), $(x, *)$ or $(*, y)$ in the set of samples (which happens with probability 0), we know absolutely nothing about the label of (x, y), and therefore by no means we can expect f' to predict the label of (x, y) correctly — in fact, f' will always assign label 0 to such a data point.*

D.2. On truthful classifiers and hill-climbing

Below, we make a few remarks regarding the generalization bound (Theorem 3) for HC.

- Observe that the generalization gap depends polynomially on the number of subclassifiers n. Without additional restrictions, n can be as large as 2^k leading to a gap which is exponential in k. This suggests that in practice, to achieve any meaningful generalization guarantee, one has to restrict the number of subclassifiers used. In fact, we do run our algorithm on a small set of features in Section 5.
- Recall that the class of linear classifiers in the k-dimensional Euclidean space has VC dimension $k+1$. So, if we restrict all subclassifiers to be linear, and require that the number of subclassifiers n to be constant, then Theorem 3 implies that with high probability, the generalization gap is

$$\widetilde{O}\left(\sqrt{\frac{k}{m}}\right),$$

where k is the number of features, m is the number of samples, and $\widetilde{O}$ hides a logarithmic factor. Our algorithms are practicable in this kind of regime.

E. Experiments

E.1. Implementation details

In our implementation, we use Python's Scikit-learn (0.22.1) package (Pedregosa et al., 2011) of classifiers and other machine learning packages whenever possible. The categorical features in the datasets are one-hot encoded. To help ensure the convergence of gradient-based classifiers, we then standardize features by removing the mean and scaling to unit variance.

For imputation-based classifiers, we use mean/mode imputation: mean for numerical and ordinal features, and mode for categorical features. For reduced-feature-based classifiers, we default to reject if the test data point's set of available features was unseen in the training process. For classification methods involving Fayyad and Irani's MDLP discretization algorithm, we use a modified version of `Discretization-MDLPC`, licensed under GPL-3[8].

The performance of each classifier under each setting is evaluated with Nx2-fold cross-validation (Dietterich, 1998): training on 50% of the data and testing on the remaining 50%; repeat N times. To tune the parameters for the classifiers, we perform grid search over a subset of the parameter space considered by (Lessmann et al., 2015), in a 5-fold cross-validation on every training set of the (outer) Nx2-cross-validation loop.

E.2. Additional experimental results

We evaluate our methods both with and without balancing the datasets through random undersampling. This is denoted by "balanced datasets" when we undersample before training and testing, and "unbalanced datasets" when we do not. Comparing Figure 1 and 2, it appears that there is no significant difference in relative accuracy across the various methods when applied to balanced and imbalanced datasets. However, comparing, for example, Table 7 and 13, we observe that when the dataset is unbalanced, classifiers based on imputation and reduced-feature modeling, when faced with strategic reporting, tend to accept everything and yield a considerably high accuracy. Many other general issues regarding the use of accuracy as a metric on unbalanced datasets are known (Japkowicz, 2000; Chawla et al., 2003; 2004). In practice, thresholding methods are sometimes used to determine a proper threshold for binary prediction in such cases (Lessmann et al., 2015; Elkan, 2001).

Therefore, in addition to accuracy, we evaluate our approach with area under the receiver operating characteristic curve (AUC). AUC becomes a useful metric when doing imbalanced classification because often a balance of false-positive and false-negative rates is desired, and it is invariant to the threshold used in binary classification.[9] For MINCUT, its receiver operating characteristic curve is undefined because it does not output probabilistic predictions; for HILL-CLIMBING, we take the maximum of the probabilistic predictions across all applicable classifiers to be the HILL-CLIMBING classifier's probabilistic prediction for a data point, and obtain AUC from that. From Table 23 to 28, we observe that HILL-CLIMBING

[8] `Discretization-MDLPC` codebase: `https://github.com/navicto/Discretization-MDLPC`.

[9] ... although AUC's global perspective assumes implicitly that all thresholds are equally probable. This is often criticized as not plausible in credit scoring (Hand & Anagnostopoulos, 2013)

classifiers generally yield a AUC as good as, if not higher than, imputation- and reduced-feature-based classifiers on imbalanced datasets (also Figure 4). The same holds for balanced datasets too (Figure 3).

For completeness, we also include the performance of classifiers based on imputation and reduced-feature modeling, with discretization. As expected, common classifiers are generally less prone to overfitting than MINCUT, and discretizing the feature space only limits their performance.

Figure 2. Selected classifier accuracy w/ strategic behavior, unbalanced datasets

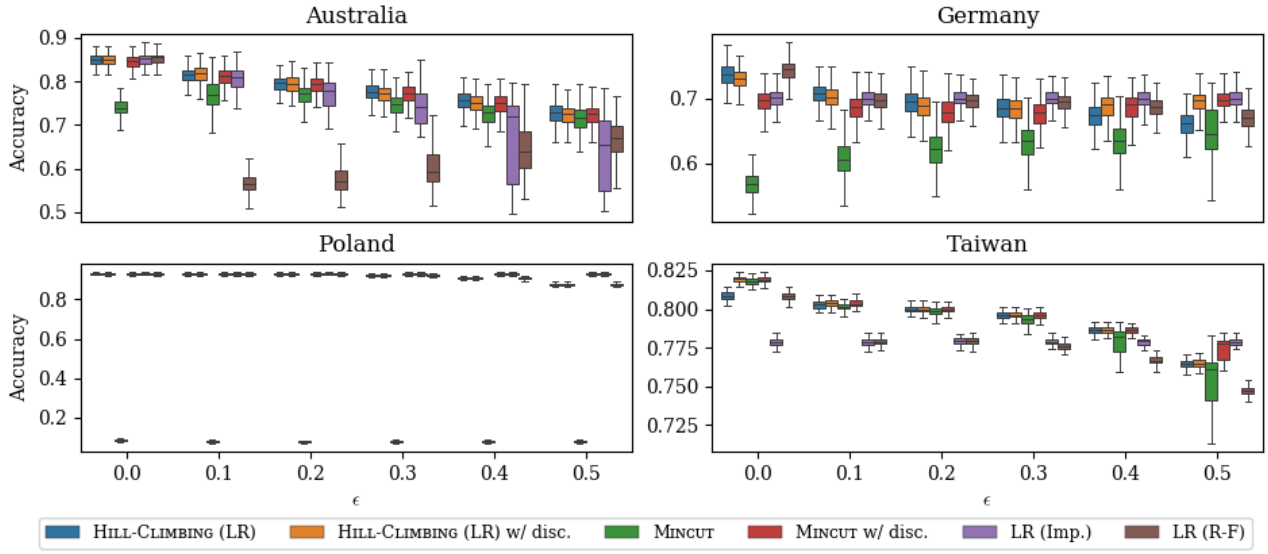

Figure 3. Selected classifier AUC w/ strategic behavior, balanced datasets

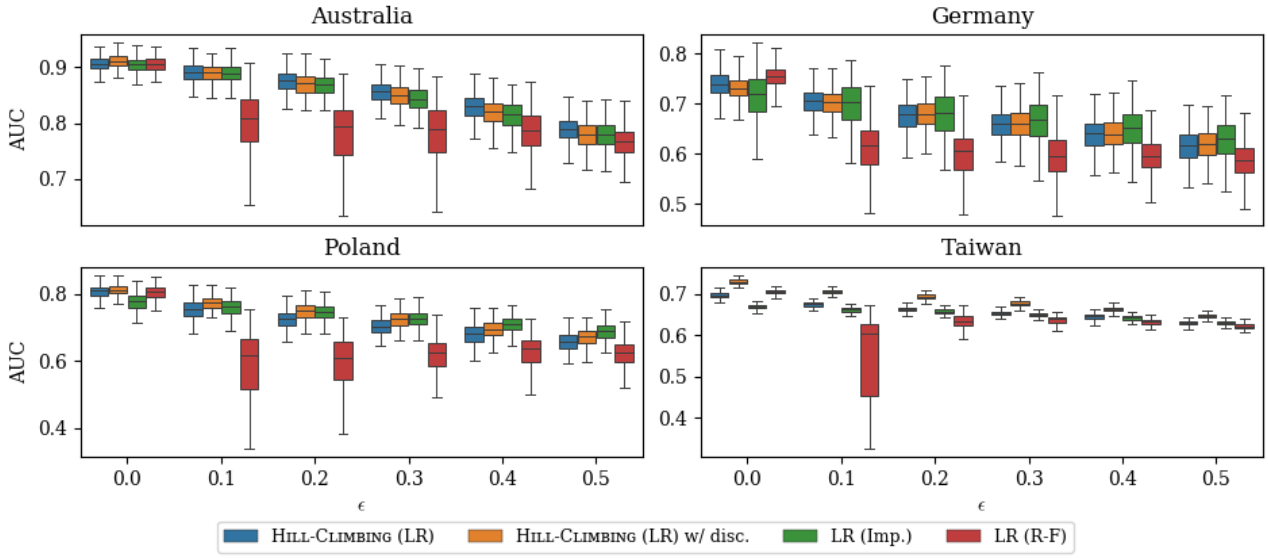

Figure 4. Selected classifier AUC w/ strategic behavior, unbalanced datasets

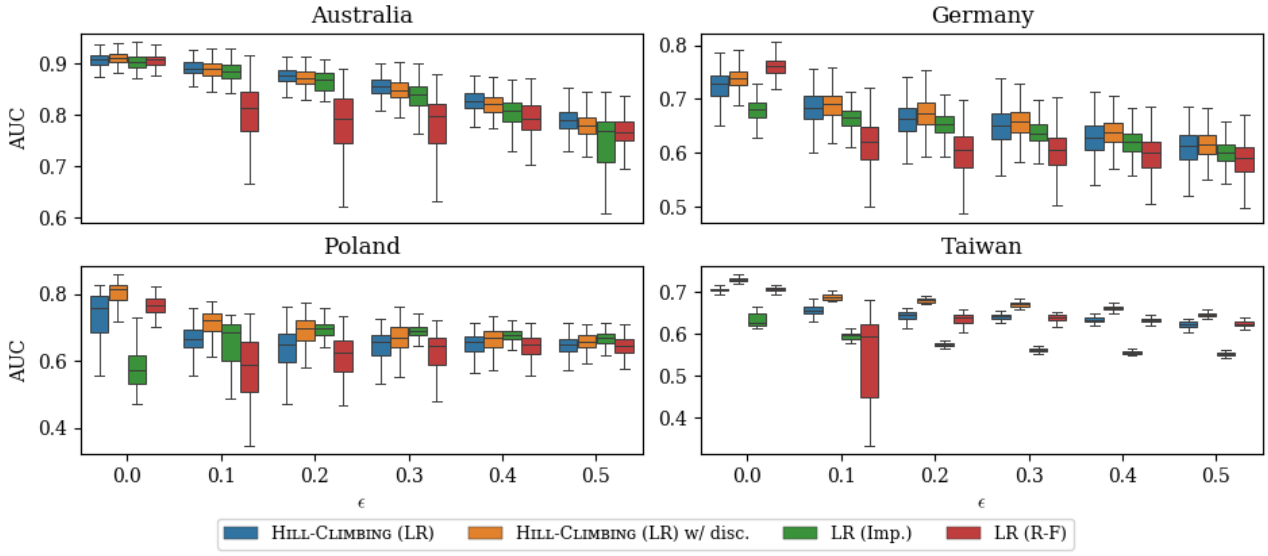

Table 5. Our methods vs. the rest: mean classifier accuracy for $\epsilon = 0.0$, balanced datasets ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.858	.858	.686	.686	.739	.739	.685	.685
HC (LR) w/ disc.	.856	.856	.679	.679	.753	.753	.695	.695
HC (ANN)	.859	.859	.692	.692	.728	.728	.688	.688
HC (ANN) w/ disc.	.849	.849	.683	.683	.754	.754	.696	.696
MINCUT	.743	.743	.576	.576	.501	.501	.700	.700
MINCUT w/ disc.	.852	.852	.658	.658	.748	.748	.701	.701
MAJ	.746	.746	.579	.579	.501	.501	.701	.701
MAJ w/ disc.	.854	.854	.653	.653	.749	.749	.701	.701
LR (Imputation)	**.861**	**.862**	.692	.599	.738	.677	.688	.637
LR (Imputation) w/ disc.	.859	.856	.682	.590	**.757**	.743	.695	.500
ANN (Imputation)	.859	.859	**.693**	.600	.729	.664	.699	.576
ANN (Imputation) w/ disc.	.849	.828	.683	.593	**.757**	.739	.697	.503
RF (Imputation)	.820	.697	.651	.561	.748	.737	.701	.531
RF (Imputation) w/ disc.	.855	.849	.668	.588	.750	.734	**.702**	.520
kNN (Imputation)	.852	.802	.672	.584	.726	.649	.674	.515
kNN (Imputation) w/ disc.	.850	.812	.669	.578	.742	.724	.675	.497
LR (Reduced Feature)	**.861**	.861	.692	.692	.738	.738	.688	.688
LR (Reduced Feature) w/ disc.	.859	.859	.682	.682	**.757**	**.757**	.695	.695
ANN (Reduced Feature)	.859	.859	**.693**	**.693**	.729	.728	.699	.699
ANN (Reduced Feature) w/ disc.	.849	.849	.684	.684	.756	.756	.697	.697
RF (Reduced Feature)	.820	.820	.651	.651	.747	.748	.701	.701
RF (Reduced Feature) w/ disc.	.855	.855	.668	.668	.750	.751	**.702**	**.702**
kNN (Reduced Feature)	.852	.852	.672	.673	.727	.727	.674	.676
kNN (Reduced Feature) w/ disc.	.848	.849	.667	.668	.736	.739	.673	.675

Table 6. Our methods vs. the rest: mean classifier accuracy for $\epsilon = 0.1$, balanced datasets ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.814	.814	.659	**.659**	.681	.681	.664	.664
HC (LR) w/ disc.	.819	.819	.656	.656	.719	**.719**	.672	.672
HC (ANN)	.803	.803	.647	.647	.664	.664	.665	.665
HC (ANN) w/ disc.	.796	.796	.634	.634	.706	.706	.671	.671
MINCUT	.770	.770	.584	.584	.502	.502	.673	**.673**
MINCUT w/ disc.	.812	.812	.641	.641	.716	.716	.672	.672
MAJ	.689	.725	.549	.536	.502	.502	.686	.565
MAJ w/ disc.	.816	.679	.619	.544	.728	.601	.693	.564
LR (Imputation)	.821	**.821**	.677	.593	.725	.667	.677	.627
LR (Imputation) w/ disc.	.823	.796	.664	.594	**.736**	.675	.690	.556
ANN (Imputation)	.822	.819	**.678**	.595	.718	.654	.692	.557
ANN (Imputation) w/ disc.	.824	.772	.667	.596	.735	.678	.693	.559
RF (Imputation)	.795	.658	.635	.560	**.736**	.655	.687	.541
RF (Imputation) w/ disc.	.821	.779	.659	.594	.733	.664	**.696**	.545
kNN (Imputation)	.818	.751	.653	.579	.713	.634	.662	.552
kNN (Imputation) w/ disc.	.812	.755	.649	.580	.722	.651	.650	.538
LR (Reduced Feature)	**.828**	.540	.651	.507	.696	.506	.674	.529
LR (Reduced Feature) w/ disc.	.823	.530	.651	.512	.730	.517	.689	.530
ANN (Reduced Feature)	.825	.541	.649	.507	.696	.505	.681	.520
ANN (Reduced Feature) w/ disc.	.813	.536	.643	.513	.727	.515	.689	.524
RF (Reduced Feature)	.793	.535	.625	.506	.720	.509	.687	.528
RF (Reduced Feature) w/ disc.	.821	.538	.642	.512	.723	.516	.693	.524
kNN (Reduced Feature)	.809	.527	.623	.505	.683	.506	.662	.524
kNN (Reduced Feature) w/ disc.	.808	.524	.631	.511	.708	.514	.662	.523

Table 7. Our methods vs. the rest: mean classifier accuracy for $\epsilon = 0.2$, balanced datasets ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.793	.793	.640	.640	.659	.659	.648	.648
HC (LR) w/ disc.	.794	**.794**	.642	**.642**	.690	.690	.650	.650
HC (ANN)	.774	.774	.625	.625	.640	.640	.647	.647
HC (ANN) w/ disc.	.769	.769	.617	.617	.679	.679	.648	.648
MINCUT	.768	.768	.581	.581	.500	.500	.652	**.652**
MINCUT w/ disc.	.789	.789	.630	.630	.693	**.693**	.649	.649
MAJ	.665	.670	.539	.524	.500	.500	.678	.560
MAJ w/ disc.	.795	.618	.605	.533	.715	.565	.688	.566
LR (Imputation)	.796	.788	**.665**	.579	.715	.662	.670	.619
LR (Imputation) w/ disc.	.799	.762	.652	.577	.719	.631	.686	.541
ANN (Imputation)	.798	.786	**.665**	.581	.704	.648	.686	.535
ANN (Imputation) w/ disc.	.799	.747	.652	.577	.718	.635	.688	.542
RF (Imputation)	.771	.630	.626	.561	**.722**	.603	.679	.549
RF (Imputation) w/ disc.	.796	.744	.643	.575	.715	.620	**.689**	.542
kNN (Imputation)	.791	.722	.638	.566	.694	.626	.662	.560
kNN (Imputation) w/ disc.	.786	.704	.628	.572	.696	.606	.655	.563
LR (Reduced Feature)	**.809**	.544	.632	.508	.669	.510	.666	.592
LR (Reduced Feature) w/ disc.	.796	.542	.633	.516	.708	.522	.684	.587
ANN (Reduced Feature)	.806	.542	.629	.509	.661	.509	.668	.546
ANN (Reduced Feature) w/ disc.	.792	.544	.629	.514	.707	.521	.684	.541
RF (Reduced Feature)	.779	.537	.611	.507	.698	.518	.679	.574
RF (Reduced Feature) w/ disc.	.799	.546	.625	.515	.706	.525	.687	.585
kNN (Reduced Feature)	.781	.531	.604	.505	.655	.510	.658	.586
kNN (Reduced Feature) w/ disc.	.775	.538	.618	.512	.684	.519	.651	.587

Table 8. Our methods vs. the rest: mean classifier accuracy for $\epsilon = 0.3$, balanced datasets ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.771	**.771**	.626	.626	.642	.642	.636	.636
HC (LR) w/ disc.	.770	.770	.630	**.630**	.670	.670	.639	**.639**
HC (ANN)	.753	.753	.611	.611	.621	.621	.636	.636
HC (ANN) w/ disc.	.748	.748	.598	.598	.658	.658	.638	.638
MINCUT	.751	.751	.579	.579	.501	.501	.638	.638
MINCUT w/ disc.	.770	.770	.620	.620	.671	**.671**	.638	.638
MAJ	.651	.653	.531	.524	.501	.498	.668	.559
MAJ w/ disc.	.775	.615	.592	.535	.701	.556	.680	.561
LR (Imputation)	.773	.757	.651	.580	.700	.651	.660	.604
LR (Imputation) w/ disc.	.770	.731	.637	.589	**.709**	.602	.680	.525
ANN (Imputation)	.775	.752	**.652**	.583	.691	.635	.677	.521
ANN (Imputation) w/ disc.	.770	.716	.638	.588	.707	.608	**.682**	.528
RF (Imputation)	.749	.626	.609	.556	.706	.564	.669	.553
RF (Imputation) w/ disc.	.771	.718	.629	.584	.707	.596	.681	.535
kNN (Imputation)	.765	.664	.623	.565	.681	.611	.651	.560
kNN (Imputation) w/ disc.	.759	.674	.608	.569	.689	.586	.641	.557
LR (Reduced Feature)	**.787**	.582	.608	.512	.650	.521	.654	.612
LR (Reduced Feature) w/ disc.	.776	.575	.616	.522	.698	.550	.678	.601
ANN (Reduced Feature)	.783	.575	.606	.512	.647	.514	.656	.570
ANN (Reduced Feature) w/ disc.	.776	.593	.612	.518	.695	.545	.676	.557
RF (Reduced Feature)	.760	.578	.592	.512	.684	.542	.669	.592
RF (Reduced Feature) w/ disc.	.782	.607	.609	.523	.702	.558	.679	.599
kNN (Reduced Feature)	.768	.573	.582	.509	.647	.530	.651	.598
kNN (Reduced Feature) w/ disc.	.759	.581	.599	.521	.681	.543	.653	.599

Table 9. Our methods vs. the rest: mean classifier accuracy for $\epsilon = 0.4$, balanced datasets ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.752	**.752**	.611	.611	.626	.626	.628	.628
HC (LR) w/ disc.	.748	.748	.618	**.618**	.649	**.649**	.629	**.629**
HC (ANN)	.731	.731	.593	.593	.602	.602	.628	.628
HC (ANN) w/ disc.	.726	.726	.586	.586	.640	.640	.627	.627
MINCUT	.727	.727	.575	.575	.500	.500	.626	.626
MINCUT w/ disc.	.746	.746	.611	.611	.648	.648	.626	.626
MAJ	.650	.633	.533	.524	.500	.500	.659	.555
MAJ w/ disc.	.760	.610	.583	.535	**.690**	.563	.671	.557
LR (Imputation)	.757	.732	.634	.579	.688	.644	.648	.601
LR (Imputation) w/ disc.	.741	.689	.622	.577	.680	.576	.671	.515
ANN (Imputation)	.759	.727	**.635**	.580	.676	.624	.668	.512
ANN (Imputation) w/ disc.	.742	.676	.622	.578	.679	.582	**.672**	.518
RF (Imputation)	.728	.633	.603	.561	.685	.558	.660	.554
RF (Imputation) w/ disc.	.741	.673	.614	.575	.678	.572	.671	.517
kNN (Imputation)	.744	.630	.606	.560	.663	.576	.641	.554
kNN (Imputation) w/ disc.	.729	.639	.591	.561	.659	.571	.641	.550
LR (Reduced Feature)	**.771**	.628	.591	.524	.628	.540	.641	.615
LR (Reduced Feature) w/ disc.	.762	.642	.598	.527	.684	.579	.670	.602
ANN (Reduced Feature)	.766	.616	.587	.522	.620	.523	.643	.603
ANN (Reduced Feature) w/ disc.	.758	.636	.587	.520	.677	.560	.669	.597
RF (Reduced Feature)	.746	.622	.582	.523	.661	.560	.660	.598
RF (Reduced Feature) w/ disc.	.767	.667	.594	.527	.685	.580	.670	.602
kNN (Reduced Feature)	.754	.624	.571	.518	.629	.558	.640	.600
kNN (Reduced Feature) w/ disc.	.747	.642	.585	.528	.666	.577	.639	.601

Table 10. Our methods vs. the rest: mean classifier accuracy for $\epsilon = 0.5$, balanced datasets ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.726	**.726**	.597	.597	.612	.612	.615	.615
HC (LR) w/ disc.	.720	.720	.601	**.601**	.638	**.638**	.616	**.616**
HC (ANN)	.709	.709	.578	.578	.587	.587	.615	.615
HC (ANN) w/ disc.	.702	.702	.570	.570	.626	.626	.615	.615
MINCUT	.707	.707	.572	.572	.500	.500	.613	.613
MINCUT w/ disc.	.720	.720	.597	.597	.632	.632	.612	.612
MAJ	.649	.615	.535	.527	.500	.498	.646	.551
MAJ w/ disc.	.734	.606	.572	.533	.667	.568	.657	.552
LR (Imputation)	.734	.706	.616	.571	**.670**	.630	.636	.584
LR (Imputation) w/ disc.	.708	.641	.604	.565	.648	.562	**.659**	.507
ANN (Imputation)	.735	.699	**.617**	.570	.658	.605	.655	.505
ANN (Imputation) w/ disc.	.709	.631	.602	.563	.647	.565	**.659**	.512
RF (Imputation)	.705	.634	.588	.555	.660	.552	.647	.549
RF (Imputation) w/ disc.	.708	.633	.597	.562	.649	.560	.657	.511
kNN (Imputation)	.720	.628	.588	.552	.646	.570	.625	.545
kNN (Imputation) w/ disc.	.692	.606	.574	.543	.622	.546	.623	.546
LR (Reduced Feature)	**.744**	.660	.572	.532	.612	.559	.626	.611
LR (Reduced Feature) w/ disc.	.737	.666	.586	.525	.664	.589	.657	.604
ANN (Reduced Feature)	.735	.641	.569	.530	.599	.531	.624	.604
ANN (Reduced Feature) w/ disc.	.731	.649	.570	.515	.655	.567	.654	.596
RF (Reduced Feature)	.723	.650	.567	.534	.635	.569	.647	.600
RF (Reduced Feature) w/ disc.	.743	.682	.584	.523	.666	.584	.657	.604
kNN (Reduced Feature)	.731	.659	.555	.528	.618	.577	.627	.599
kNN (Reduced Feature) w/ disc.	.723	.668	.572	.527	.651	.585	.627	.599

Table 11. Our methods vs. the rest: mean classifier accuracy for $\epsilon = 0.0$, unbalanced datasets ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.848	.848	.736	.736	.931	.931	.808	.808
HC (LR) w/ disc.	.848	.848	.730	.730	.931	.931	.819	.819
HC (ANN)	.850	.850	.740	.740	.931	.931	.819	.819
HC (ANN) w/ disc.	.848	.848	.730	.730	.931	.931	.819	.819
MINCUT	.740	.740	.567	.567	.085	.085	.818	.818
MINCUT w/ disc.	.845	.845	.696	.696	.929	.929	.819	.819
MAJ	.738	.738	.553	.553	.085	.085	.818	.818
MAJ w/ disc.	.847	.847	.686	.686	.928	.928	.819	.819
LR (Imputation)	**.853**	.821	.744	.701	.930	.931	.808	.779
LR (Imputation) w/ disc.	.850	.797	.731	.700	**.932**	.931	**.820**	.779
ANN (Imputation)	.849	.816	.744	.701	.930	.931	.819	.779
ANN (Imputation) w/ disc.	.848	.774	.730	.700	**.932**	.931	**.820**	.779
RF (Imputation)	.818	.635	.705	.701	.929	.931	.818	.779
RF (Imputation) w/ disc.	.846	.790	.719	.700	.930	.931	.819	.779
kNN (Imputation)	.845	.745	.727	.701	.931	.931	.810	.775
kNN (Imputation) w/ disc.	.840	.747	.716	.699	.930	.920	.811	.769
LR (Reduced Feature)	**.853**	**.853**	.744	**.744**	.930	.930	.808	.808
LR (Reduced Feature) w/ disc.	.850	.850	.731	.731	**.932**	**.932**	**.820**	**.820**
ANN (Reduced Feature)	.850	.850	**.745**	**.744**	.930	.930	.819	.819
ANN (Reduced Feature) w/ disc.	.848	.848	.729	.730	.931	.931	**.820**	**.820**
RF (Reduced Feature)	.818	.818	.705	.706	.929	.929	.818	.818
RF (Reduced Feature) w/ disc.	.846	.847	.719	.719	.930	.930	.819	.819
kNN (Reduced Feature)	.845	.845	.728	.728	.931	.931	.811	.810
kNN (Reduced Feature) w/ disc.	.843	.841	.717	.717	.930	.931	.810	.807

Table 12. Our methods vs. the rest: mean classifier accuracy for $\epsilon = 0.1$, unbalanced datasets ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.814	.814	.708	**.708**	**.930**	**.930**	.803	.803
HC (LR) w/ disc.	.818	**.818**	.701	.701	.929	.929	.804	**.804**
HC (ANN)	.799	.799	.699	.699	**.930**	**.930**	.803	.803
HC (ANN) w/ disc.	.797	.797	.693	.693	**.930**	**.930**	.802	.802
MINCUT	.769	.769	.606	.606	.080	.080	.802	.802
MINCUT w/ disc.	.811	.811	.686	.686	.928	.928	.804	**.804**
MAJ	.677	.734	.489	.637	.077	.145	.811	.782
MAJ w/ disc.	.814	.703	.643	.686	.922	.929	.815	.782
LR (Imputation)	.818	.777	**.735**	.701	**.930**	**.930**	.805	.779
LR (Imputation) w/ disc.	.826	.747	.723	.700	**.930**	.929	.816	.779
ANN (Imputation)	.822	.771	**.735**	.701	**.930**	**.930**	.815	.779
ANN (Imputation) w/ disc.	**.827**	.737	.722	.700	**.930**	.929	**.817**	.779
RF (Imputation)	.794	.640	.696	.700	.928	**.930**	.813	.779
RF (Imputation) w/ disc.	.823	.735	.708	.700	.928	.929	**.817**	.779
kNN (Imputation)	.813	.704	.719	.700	**.930**	**.930**	.808	.779
kNN (Imputation) w/ disc.	.814	.698	.707	.696	.927	.929	.808	.779
LR (Reduced Feature)	.825	.570	.715	.697	.929	**.930**	.806	.779
LR (Reduced Feature) w/ disc.	.818	.572	.713	.697	.929	.929	**.817**	.779
ANN (Reduced Feature)	.823	.571	.715	.697	**.930**	**.930**	.811	.779
ANN (Reduced Feature) w/ disc.	.818	.577	.711	.697	.929	.929	.814	.779
RF (Reduced Feature)	.795	.570	.687	.697	.927	**.930**	.813	.779
RF (Reduced Feature) w/ disc.	.817	.578	.698	.697	.926	.929	.816	.779
kNN (Reduced Feature)	.807	.563	.701	.697	**.930**	**.930**	.808	.779
kNN (Reduced Feature) w/ disc.	.804	.565	.699	.697	.927	.929	.809	.779

Table 13. Our methods vs. the rest: mean classifier accuracy for $\epsilon = 0.2$, unbalanced datasets ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.796	**.796**	.694	.694	.929	.929	.800	**.800**
HC (LR) w/ disc.	.794	.794	.688	.688	.929	.929	.800	**.800**
HC (ANN)	.779	.779	.682	.682	.930	.930	.800	**.800**
HC (ANN) w/ disc.	.771	.771	.686	.686	.930	.930	.800	**.800**
MINCUT	.769	.769	.621	.621	.078	.078	.798	.798
MINCUT w/ disc.	.794	.794	.678	.678	.929	.929	.800	**.800**
MAJ	.652	.700	.471	.678	.074	.565	.809	.780
MAJ w/ disc.	.798	.648	.631	.694	.922	**.931**	**.814**	.781
LR (Imputation)	.803	.742	**.728**	**.700**	.930	**.931**	.802	.779
LR (Imputation) w/ disc.	.799	.711	.718	**.700**	.930	.930	.813	.779
ANN (Imputation)	.806	.731	.727	**.700**	**.931**	**.931**	.813	.779
ANN (Imputation) w/ disc.	.798	.700	.716	**.700**	.930	.930	**.814**	.779
RF (Imputation)	.775	.627	.690	**.700**	.928	**.931**	.811	.780
RF (Imputation) w/ disc.	.797	.693	.706	**.700**	.929	.930	**.814**	.779
kNN (Imputation)	.792	.678	.711	.699	.930	**.931**	.807	.780
kNN (Imputation) w/ disc.	.789	.672	.698	.695	.928	.930	.801	.778
LR (Reduced Feature)	**.807**	.577	.701	.696	.929	.930	.803	.779
LR (Reduced Feature) w/ disc.	.797	.567	.705	.698	.928	.929	.813	.780
ANN (Reduced Feature)	.805	.576	.699	.696	.930	.930	.806	.778
ANN (Reduced Feature) w/ disc.	.794	.576	.701	.698	.929	.929	.809	.778
RF (Reduced Feature)	.780	.578	.675	.696	.924	.928	.810	.782
RF (Reduced Feature) w/ disc.	.798	.577	.689	.698	.925	.929	.813	.785
kNN (Reduced Feature)	.782	.567	.691	.696	.929	.929	.806	.781
kNN (Reduced Feature) w/ disc.	.780	.566	.693	.697	.928	.929	.806	.782

Table 14. Our methods vs. the rest: mean classifier accuracy for $\epsilon = 0.3$, unbalanced datasets ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.775	**.775**	.685	.685	.923	.923	.796	**.796**
HC (LR) w/ disc.	.771	.771	.682	.682	.923	.923	.796	**.796**
HC (ANN)	.755	.755	.673	.673	.923	.923	.796	**.796**
HC (ANN) w/ disc.	.747	.747	.692	.692	.923	.923	.796	**.796**
MINCUT	.746	.746	.634	.634	.079	.079	.793	.793
MINCUT w/ disc.	.772	.772	.676	.676	.930	**.930**	.796	**.796**
MAJ	.642	.659	.476	.692	.079	.728	.807	.780
MAJ w/ disc.	.784	.630	.631	.698	.923	**.930**	.811	.781
LR (Imputation)	.783	.709	**.723**	**.701**	.930	**.930**	.799	.779
LR (Imputation) w/ disc.	.773	.672	.710	.699	.930	**.930**	.809	.779
ANN (Imputation)	.784	.695	.722	**.701**	.930	**.930**	.808	.779
ANN (Imputation) w/ disc.	.772	.661	.709	.699	**.931**	**.930**	**.812**	.779
RF (Imputation)	.754	.616	.688	**.701**	.926	**.930**	.809	.780
RF (Imputation) w/ disc.	.774	.659	.700	.699	.929	**.930**	.811	.779
kNN (Imputation)	.770	.636	.704	.697	.930	**.930**	.804	.779
kNN (Imputation) w/ disc.	.763	.640	.693	.688	.928	.927	.801	.775
LR (Reduced Feature)	**.791**	.604	.688	.693	.923	.923	.797	.776
LR (Reduced Feature) w/ disc.	.783	.595	.697	.696	.923	.923	.807	.780
ANN (Reduced Feature)	.788	.601	.689	.693	.923	.923	.797	.775
ANN (Reduced Feature) w/ disc.	.780	.615	.696	.696	.923	.923	.802	.774
RF (Reduced Feature)	.765	.610	.666	.692	.915	.920	.804	.782
RF (Reduced Feature) w/ disc.	.784	.632	.683	.695	.920	.923	.806	.783
kNN (Reduced Feature)	.771	.598	.685	.692	.923	.923	.800	.780
kNN (Reduced Feature) w/ disc.	.768	.600	.686	.695	.922	.923	.798	.777

Table 15. Our methods vs. the rest: mean classifier accuracy for $\epsilon = 0.4$, unbalanced datasets ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.756	**.756**	.673	.673	.908	.908	.786	.786
HC (LR) w/ disc.	.750	.750	.686	.686	.908	.908	.787	**.787**
HC (ANN)	.733	.733	.669	.669	.909	.909	.787	**.787**
HC (ANN) w/ disc.	.725	.725	.694	.694	.909	.909	.787	**.787**
MINCUT	.725	.725	.633	.633	.079	.079	.778	.778
MINCUT w/ disc.	.748	.748	.685	.685	**.931**	**.931**	.785	.785
MAJ	.645	.632	.500	.697	.093	.823	.803	.779
MAJ w/ disc.	.767	.618	.640	.699	.924	**.931**	.808	.780
LR (Imputation)	.763	.677	**.716**	.700	**.931**	**.931**	.796	.779
LR (Imputation) w/ disc.	.749	.636	.708	**.701**	.929	.930	.807	.779
ANN (Imputation)	.763	.657	.715	.700	**.931**	**.931**	.804	.779
ANN (Imputation) w/ disc.	.748	.628	.707	**.701**	.930	.930	**.809**	.779
RF (Imputation)	.736	.603	.685	.700	.924	**.931**	.805	.779
RF (Imputation) w/ disc.	.749	.626	.700	**.701**	.929	.930	.808	.780
kNN (Imputation)	.748	.616	.696	.692	.930	**.931**	.801	.779
kNN (Imputation) w/ disc.	.735	.627	.687	.682	.926	.930	.800	.761
LR (Reduced Feature)	**.775**	.644	.675	.686	.908	.909	.785	.767
LR (Reduced Feature) w/ disc.	.764	.631	.694	.697	.908	.908	.795	.777
ANN (Reduced Feature)	.768	.629	.675	.686	.909	.909	.782	.765
ANN (Reduced Feature) w/ disc.	.761	.639	.694	.697	.908	.908	.790	.767
RF (Reduced Feature)	.750	.648	.654	.682	.896	.901	.791	.775
RF (Reduced Feature) w/ disc.	.770	.670	.683	.696	.905	.908	.794	.778
kNN (Reduced Feature)	.755	.647	.674	.684	.908	.908	.787	.772
kNN (Reduced Feature) w/ disc.	.755	.649	.681	.695	.907	.908	.787	.772

Table 16. Our methods vs. the rest: mean classifier accuracy for $\epsilon = 0.5$, unbalanced datasets ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.729	**.729**	.660	.660	.877	.877	.764	.764
HC (LR) w/ disc.	.723	.723	.692	.692	.879	.879	.765	.765
HC (ANN)	.713	.713	.663	.663	.877	.877	.765	.765
HC (ANN) w/ disc.	.705	.705	.694	.694	.879	.879	.765	.765
MINCUT	.714	.714	.647	.647	.095	.095	.752	.752
MINCUT w/ disc.	.724	.724	.696	.696	.930	.930	.770	.770
MAJ	.643	.615	.531	.699	.125	.879	.800	.779
MAJ w/ disc.	.744	.608	.649	.699	.924	**.931**	**.805**	**.780**
LR (Imputation)	.736	.628	**.710**	.700	.930	**.931**	.793	.779
LR (Imputation) w/ disc.	.719	.601	.703	**.701**	.930	**.931**	.804	.779
ANN (Imputation)	.734	.614	.709	.700	.930	**.931**	.802	.779
ANN (Imputation) w/ disc.	.720	.595	.702	**.701**	**.931**	**.931**	.804	.779
RF (Imputation)	.711	.592	.683	.700	.920	**.931**	.803	.779
RF (Imputation) w/ disc.	.721	.598	.698	**.701**	.930	**.931**	.804	.779
kNN (Imputation)	.723	.602	.688	.690	.929	.929	.797	.777
kNN (Imputation) w/ disc.	.702	.597	.677	.679	.929	**.931**	.796	.774
LR (Reduced Feature)	**.749**	.668	.657	.670	.876	.877	.761	.747
LR (Reduced Feature) w/ disc.	.744	.669	.689	.696	.877	.877	.770	.758
ANN (Reduced Feature)	.740	.648	.657	.670	.877	.877	.756	.746
ANN (Reduced Feature) w/ disc.	.735	.651	.692	.696	.877	.877	.765	.748
RF (Reduced Feature)	.729	.668	.639	.665	.861	.865	.768	.757
RF (Reduced Feature) w/ disc.	.748	.689	.678	.695	.875	.877	.770	.759
kNN (Reduced Feature)	.732	.670	.654	.666	.876	.877	.764	.751
kNN (Reduced Feature) w/ disc.	.730	.677	.673	.689	.876	.877	.765	.753

Table 17. Our methods vs. the rest: mean classifier AUC for $\epsilon = 0.0$, balanced datasets ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.906	.906	.739	.739	.807	.807	.696	.696
HC (LR) w/ disc.	**.911**	**.911**	.732	.732	.813	.813	.729	.729
HC (ANN)	.908	.908	.753	.753	.799	.799	.701	.701
HC (ANN) w/ disc.	.910	.910	.735	.735	.815	.815	.731	.731
MINCUT	-	-	-	-	-	-	-	-
MINCUT w/ disc.	-	-	-	-	-	-	-	-
MAJ	-	-	-	-	-	-	-	-
MAJ w/ disc.	-	-	-	-	-	-	-	-
LR (Imputation)	.906	.905	.756	.717	.805	.776	.704	.667
LR (Imputation) w/ disc.	.910	.897	.739	.686	**.819**	.796	.731	.587
ANN (Imputation)	.907	.905	**.757**	.717	.801	.771	.732	.683
ANN (Imputation) w/ disc.	.910	.897	.742	.690	.818	.795	.731	.587
RF (Imputation)	.885	.876	.709	.659	**.819**	.811	**.734**	.582
RF (Imputation) w/ disc.	**.911**	.899	.712	.665	.811	.790	.731	.529
kNN (Imputation)	.905	.874	.717	.663	.794	.772	.716	.636
kNN (Imputation) w/ disc.	.903	.886	.715	.651	.804	.782	.713	.525
LR (Reduced Feature)	.906	.906	.756	.756	.805	.805	.704	.704
LR (Reduced Feature) w/ disc.	.910	.910	.739	.739	**.819**	**.819**	.731	.731
ANN (Reduced Feature)	.907	.907	**.757**	**.757**	.801	.801	.731	.732
ANN (Reduced Feature) w/ disc.	.910	.910	.742	.742	.818	.818	.731	.731
RF (Reduced Feature)	.885	.885	.709	.708	**.819**	**.819**	**.734**	**.734**
RF (Reduced Feature) w/ disc.	**.911**	**.911**	.712	.712	.811	.811	.731	.731
kNN (Reduced Feature)	.905	.905	.718	.718	.795	.794	.716	.716
kNN (Reduced Feature) w/ disc.	.903	.903	.714	.715	.803	.804	.713	.715

Table 18. Our methods vs. the rest: mean classifier AUC for $\epsilon = 0.1$, balanced datasets ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.889	**.889**	.704	.704	.752	.752	.674	.674
HC (LR) w/ disc.	.889	**.889**	.702	.702	.774	**.774**	.705	.705
HC (ANN)	.886	.886	.708	**.708**	.731	.731	.682	.682
HC (ANN) w/ disc.	.877	.877	.681	.681	.766	.766	.706	**.706**
MINCUT	-	-	-	-	-	-	-	-
MINCUT w/ disc.	-	-	-	-	-	-	-	-
MAJ	-	-	-	-	-	-	-	-
MAJ w/ disc.	-	-	-	-	-	-	-	-
LR (Imputation)	.893	.888	.738	.701	.792	.759	.695	.661
LR (Imputation) w/ disc.	**.894**	.860	.720	.676	.800	.755	.725	.677
ANN (Imputation)	**.894**	.888	**.739**	.702	.787	.753	.722	.678
ANN (Imputation) w/ disc.	.893	.860	.724	.679	.800	.757	**.726**	.677
RF (Imputation)	.866	.850	.688	.656	**.806**	**.774**	.717	.633
RF (Imputation) w/ disc.	**.894**	.857	.695	.660	.791	.739	.725	.663
kNN (Imputation)	.886	.860	.699	.659	.778	.748	.708	.664
kNN (Imputation) w/ disc.	.886	.849	.691	.640	.780	.719	.701	.596
LR (Reduced Feature)	.881	.798	.697	.610	.758	.588	.693	.548
LR (Reduced Feature) w/ disc.	.885	.786	.701	.618	.790	.658	.724	.582
ANN (Reduced Feature)	.878	.820	.688	.626	.758	.633	.706	.589
ANN (Reduced Feature) w/ disc.	.876	.818	.688	.630	.786	.695	.724	.632
RF (Reduced Feature)	.860	.799	.674	.612	.789	.664	.717	.595
RF (Reduced Feature) w/ disc.	.880	.804	.680	.631	.777	.683	.722	.578
kNN (Reduced Feature)	.871	.795	.663	.599	.744	.651	.707	.647
kNN (Reduced Feature) w/ disc.	.872	.788	.677	.616	.767	.690	.703	.638

Table 19. Our methods vs. the rest: mean classifier AUC for $\epsilon = 0.2$, balanced datasets ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.874	**.874**	.677	.677	.726	.726	.662	.662
HC (LR) w/ disc.	.870	.870	.679	.679	.747	**.747**	.693	**.693**
HC (ANN)	.865	.865	.675	.675	.701	.701	.670	.670
HC (ANN) w/ disc.	.849	.849	.654	.654	.735	.735	.691	.691
MINCUT	-	-	-	-	-	-	-	-
MINCUT w/ disc.	-	-	-	-	-	-	-	-
MAJ	-	-	-	-	-	-	-	-
MAJ w/ disc.	-	-	-	-	-	-	-	-
LR (Imputation)	**.879**	.868	.719	.680	.781	.745	.690	.657
LR (Imputation) w/ disc.	.876	.833	.700	.656	.780	.712	**.721**	.682
ANN (Imputation)	**.879**	.868	**.721**	**.681**	.774	.738	.714	.673
ANN (Imputation) w/ disc.	.875	.832	.704	.658	.778	.713	**.721**	.682
RF (Imputation)	.850	.823	.669	.641	**.788**	.733	.709	.638
RF (Imputation) w/ disc.	.875	.831	.679	.643	.771	.702	.720	.671
kNN (Imputation)	.869	.842	.681	.640	.757	.726	.702	.655
kNN (Imputation) w/ disc.	.863	.816	.670	.628	.756	.676	.698	.619
LR (Reduced Feature)	.866	.778	.668	.600	.726	.591	.687	.630
LR (Reduced Feature) w/ disc.	.863	.750	.678	.615	.765	.632	.717	.664
ANN (Reduced Feature)	.863	.804	.662	.609	.724	.639	.694	.635
ANN (Reduced Feature) w/ disc.	.857	.796	.670	.623	.760	.677	.717	.666
RF (Reduced Feature)	.846	.777	.651	.593	.762	.645	.708	.646
RF (Reduced Feature) w/ disc.	.857	.787	.661	.629	.755	.660	.716	.657
kNN (Reduced Feature)	.845	.749	.636	.568	.706	.622	.700	.659
kNN (Reduced Feature) w/ disc.	.840	.735	.656	.597	.741	.657	.696	.660

Table 20. Our methods vs. the rest: mean classifier AUC for $\epsilon = 0.3$, balanced datasets ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.855	**.855**	.658	.658	.704	.704	.653	.653
HC (LR) w/ disc.	.848	.848	.660	.660	.724	.724	.676	.676
HC (ANN)	.843	.843	.652	.652	.674	.674	.657	.657
HC (ANN) w/ disc.	.826	.826	.628	.628	.709	.709	.679	**.679**
MINCUT	-	-	-	-	-	-	-	-
MINCUT w/ disc.	-	-	-	-	-	-	-	-
MAJ	-	-	-	-	-	-	-	-
MAJ w/ disc.	-	-	-	-	-	-	-	-
LR (Imputation)	.861	.842	.704	.667	.768	**.727**	.685	.649
LR (Imputation) w/ disc.	.855	.793	.683	.648	.765	.682	**.716**	.674
ANN (Imputation)	**.862**	.842	**.706**	**.668**	.761	.720	.704	.663
ANN (Imputation) w/ disc.	.853	.793	.685	.649	.762	.681	**.716**	.675
RF (Imputation)	.828	.794	.650	.629	**.770**	.694	.700	.652
RF (Imputation) w/ disc.	.854	.793	.663	.637	.760	.679	.714	.671
kNN (Imputation)	.849	.809	.664	.623	.740	.702	.692	.641
kNN (Imputation) w/ disc.	.841	.776	.647	.618	.739	.652	.686	.618
LR (Reduced Feature)	.848	.775	.643	.595	.704	.609	.679	.636
LR (Reduced Feature) w/ disc.	.843	.749	.658	.611	.752	.651	.709	.670
ANN (Reduced Feature)	.844	.791	.640	.599	.707	.638	.682	.634
ANN (Reduced Feature) w/ disc.	.839	.785	.650	.613	.749	.672	.708	.667
RF (Reduced Feature)	.828	.756	.628	.588	.739	.647	.697	.652
RF (Reduced Feature) w/ disc.	.841	.776	.647	.623	.746	.666	.708	.669
kNN (Reduced Feature)	.825	.742	.610	.562	.691	.628	.690	.649
kNN (Reduced Feature) w/ disc.	.819	.730	.632	.596	.728	.651	.689	.650

Table 21. Our methods vs. the rest: mean classifier AUC for $\epsilon = 0.4$, balanced datasets ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.828	**.828**	.640	.640	.681	.681	.644	.644
HC (LR) w/ disc.	.819	.819	.641	.641	.694	.694	.663	.663
HC (ANN)	.809	.809	.626	.626	.647	.647	.647	.647
HC (ANN) w/ disc.	.794	.794	.604	.604	.684	.684	.664	**.664**
MINCUT	-	-	-	-	-	-	-	-
MINCUT w/ disc.	-	-	-	-	-	-	-	-
MAJ	-	-	-	-	-	-	-	-
MAJ w/ disc.	-	-	-	-	-	-	-	-
LR (Imputation)	**.844**	.813	.686	**.652**	**.754**	**.709**	.680	.641
LR (Imputation) w/ disc.	.828	.761	.664	.629	.737	.645	**.707**	.661
ANN (Imputation)	**.844**	.813	**.688**	**.652**	.745	.704	.696	.652
ANN (Imputation) w/ disc.	.828	.761	.665	.629	.735	.644	**.707**	.661
RF (Imputation)	.808	.766	.640	.619	.746	.679	.692	.650
RF (Imputation) w/ disc.	.829	.761	.646	.619	.732	.643	.705	.658
kNN (Imputation)	.827	.775	.642	.611	.719	.675	.682	.625
kNN (Imputation) w/ disc.	.816	.741	.624	.599	.718	.622	.679	.610
LR (Reduced Feature)	.825	.780	.622	.596	.677	.623	.667	.631
LR (Reduced Feature) w/ disc.	.821	.766	.638	.611	.732	.672	.694	.661
ANN (Reduced Feature)	.820	.780	.618	.594	.676	.624	.667	.633
ANN (Reduced Feature) w/ disc.	.818	.778	.630	.603	.729	.671	.693	.661
RF (Reduced Feature)	.804	.751	.613	.593	.705	.647	.684	.649
RF (Reduced Feature) w/ disc.	.820	.782	.630	.617	.729	.676	.693	.661
kNN (Reduced Feature)	.805	.745	.591	.570	.666	.634	.676	.639
kNN (Reduced Feature) w/ disc.	.803	.746	.616	.593	.710	.661	.675	.637

Table 22. Our methods vs. the rest: mean classifier AUC for $\epsilon = 0.5$, balanced datasets ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.788	**.788**	.616	.616	.657	.657	.630	.630
HC (LR) w/ disc.	.780	.780	.618	.618	.671	.671	.645	**.645**
HC (ANN)	.773	.773	.600	.600	.624	.624	.632	.632
HC (ANN) w/ disc.	.762	.762	.581	.581	.662	.662	.645	**.645**
MINCUT	-	-	-	-	-	-	-	-
MINCUT w/ disc.	-	-	-	-	-	-	-	-
MAJ	-	-	-	-	-	-	-	-
MAJ w/ disc.	-	-	-	-	-	-	-	-
LR (Imputation)	**.819**	.779	.663	.628	**.735**	**.687**	.672	.630
LR (Imputation) w/ disc.	.791	.711	.642	.607	.699	.613	**.695**	.643
ANN (Imputation)	**.819**	.779	**.665**	**.629**	.728	.682	.684	.637
ANN (Imputation) w/ disc.	.790	.710	.639	.605	.695	.613	**.695**	.643
RF (Imputation)	.782	.735	.622	.602	.720	.656	.681	.640
RF (Imputation) w/ disc.	.791	.710	.628	.601	.697	.613	.694	.642
kNN (Imputation)	.797	.744	.620	.593	.703	.652	.667	.609
kNN (Imputation) w/ disc.	.774	.693	.605	.578	.673	.585	.664	.589
LR (Reduced Feature)	.787	.765	.594	.586	.652	.621	.649	.621
LR (Reduced Feature) w/ disc.	.779	.750	.623	.598	.705	.658	.672	**.645**
ANN (Reduced Feature)	.781	.755	.590	.580	.647	.609	.647	.621
ANN (Reduced Feature) w/ disc.	.777	.751	.607	.579	.701	.652	.670	.642
RF (Reduced Feature)	.768	.730	.589	.582	.669	.635	.663	.636
RF (Reduced Feature) w/ disc.	.786	.759	.619	.600	.704	.660	.672	.644
kNN (Reduced Feature)	.770	.738	.570	.567	.646	.632	.656	.624
kNN (Reduced Feature) w/ disc.	.765	.739	.598	.583	.687	.639	.655	.624

Table 23. Our methods vs. the rest: mean classifier AUC for $\epsilon = 0.0$, unbalanced datasets ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.907	.907	.723	.723	.728	.728	.705	.705
HC (LR) w/ disc.	**.911**	**.911**	.739	.739	.798	.798	.730	.730
HC (ANN)	.908	.908	.746	.746	.804	.804	.731	.731
HC (ANN) w/ disc.	.910	.910	.747	.747	.818	.818	.730	.730
MINCUT	-	-	-	-	-	-	-	-
MINCUT w/ disc.	-	-	-	-	-	-	-	-
MAJ	-	-	-	-	-	-	-	-
MAJ w/ disc.	-	-	-	-	-	-	-	-
LR (Imputation)	.907	.895	.761	.679	.766	.579	.705	.634
LR (Imputation) w/ disc.	.910	.866	.744	.643	.826	.620	.731	.587
ANN (Imputation)	.908	.895	.762	.680	.809	.728	**.737**	.654
ANN (Imputation) w/ disc.	.910	.866	.747	.646	**.827**	.622	.732	.587
RF (Imputation)	.886	.841	.711	.579	.814	.704	.736	.541
RF (Imputation) w/ disc.	**.911**	.822	.717	.604	.806	.550	.732	.523
kNN (Imputation)	.906	.831	.725	.567	.758	.530	.709	.535
kNN (Imputation) w/ disc.	.905	.825	.710	.567	.761	.491	.706	.491
LR (Reduced Feature)	.907	.907	.761	.761	.766	.766	.705	.705
LR (Reduced Feature) w/ disc.	.910	.910	.744	.744	.826	.826	.731	.731
ANN (Reduced Feature)	.908	.908	**.763**	**.762**	.810	.810	**.737**	**.736**
ANN (Reduced Feature) w/ disc.	.910	.910	.747	.747	**.827**	**.827**	.732	.732
RF (Reduced Feature)	.886	.886	.711	.711	.814	.814	.736	**.736**
RF (Reduced Feature) w/ disc.	**.911**	**.911**	.717	.717	.806	.807	.732	.732
kNN (Reduced Feature)	.906	.906	.724	.725	.759	.758	.709	.710
kNN (Reduced Feature) w/ disc.	.905	.904	.709	.710	.763	.762	.706	.708

Table 24. Our methods vs. the rest: mean classifier AUC for $\epsilon = 0.1$, unbalanced datasets ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.891	**.891**	.683	.683	.651	.651	.655	.655
HC (LR) w/ disc.	.890	.890	.687	.687	.713	.713	.680	.680
HC (ANN)	.886	.886	.691	**.691**	.684	.684	.673	.673
HC (ANN) w/ disc.	.876	.876	.671	.671	.742	**.742**	.687	**.687**
MINCUT	-	-	-	-	-	-	-	-
MINCUT w/ disc.	-	-	-	-	-	-	-	-
MAJ	-	-	-	-	-	-	-	-
MAJ w/ disc.	-	-	-	-	-	-	-	-
LR (Imputation)	.893	.874	.741	.666	.760	.658	.694	.596
LR (Imputation) w/ disc.	**.894**	.834	.723	.633	.806	.643	.732	.678
ANN (Imputation)	.893	.874	**.743**	.667	.802	.722	.724	.599
ANN (Imputation) w/ disc.	.892	.835	.726	.635	**.809**	.643	**.733**	.678
RF (Imputation)	.865	.828	.685	.592	.799	.654	.720	.626
RF (Imputation) w/ disc.	.893	.821	.696	.605	.793	.595	**.733**	.673
kNN (Imputation)	.886	.829	.703	.586	.742	.539	.704	.632
kNN (Imputation) w/ disc.	.883	.800	.682	.575	.749	.517	.703	.576
LR (Reduced Feature)	.885	.799	.709	.616	.749	.580	.696	.539
LR (Reduced Feature) w/ disc.	.886	.788	.710	.638	.797	.708	.725	.565
ANN (Reduced Feature)	.883	.826	.707	.627	.771	.685	.714	.607
ANN (Reduced Feature) w/ disc.	.883	.827	.704	.641	.786	.739	.726	.671
RF (Reduced Feature)	.863	.798	.679	.621	.783	.554	.720	.565
RF (Reduced Feature) w/ disc.	.880	.802	.685	.637	.756	.605	.724	.539
kNN (Reduced Feature)	.874	.796	.678	.616	.727	.529	.703	.633
kNN (Reduced Feature) w/ disc.	.872	.795	.681	.628	.738	.537	.701	.619

Table 25. Our methods vs. the rest: mean classifier AUC for $\epsilon = 0.2$, unbalanced datasets ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.876	**.876**	.662	.662	.627	.627	.642	.642
HC (LR) w/ disc.	.872	.872	.669	**.669**	.683	.683	.676	.676
HC (ANN)	.866	.866	.655	.655	.677	.677	.654	.654
HC (ANN) w/ disc.	.854	.854	.639	.639	.703	.703	.679	.679
MINCUT	-	-	-	-	-	-	-	-
MINCUT w/ disc.	-	-	-	-	-	-	-	-
MAJ	-	-	-	-	-	-	-	-
MAJ w/ disc.	-	-	-	-	-	-	-	-
LR (Imputation)	**.880**	.854	.722	.653	.758	.684	.684	.577
LR (Imputation) w/ disc.	.874	.798	.704	.621	.796	.653	.724	.670
ANN (Imputation)	**.880**	.854	**.724**	.654	.790	**.711**	.710	.579
ANN (Imputation) w/ disc.	.873	.798	.707	.621	**.799**	.651	**.726**	.670
RF (Imputation)	.850	.794	.665	.603	.781	.598	.711	.639
RF (Imputation) w/ disc.	.873	.795	.681	.603	.786	.631	.725	.666
kNN (Imputation)	.870	.812	.681	.587	.728	.538	.697	.634
kNN (Imputation) w/ disc.	.862	.773	.661	.572	.737	.523	.692	.576
LR (Reduced Feature)	.870	.781	.680	.600	.739	.610	.690	.633
LR (Reduced Feature) w/ disc.	.866	.759	.689	.619	.783	.659	.718	.671
ANN (Reduced Feature)	.867	.812	.674	.611	.752	.674	.699	.656
ANN (Reduced Feature) w/ disc.	.861	.801	.678	.618	.776	.703	.718	**.685**
RF (Reduced Feature)	.850	.771	.656	.596	.757	.526	.710	.643
RF (Reduced Feature) w/ disc.	.860	.782	.667	.621	.738	.585	.718	.662
kNN (Reduced Feature)	.846	.742	.649	.577	.711	.535	.696	.646
kNN (Reduced Feature) w/ disc.	.845	.742	.660	.599	.726	.522	.693	.639

Table 26. Our methods vs. the rest: mean classifier AUC for $\epsilon = 0.3$, unbalanced datasets ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.855	**.855**	.648	.648	.639	.639	.641	.641
HC (LR) w/ disc.	.849	.849	.655	**.655**	.653	.653	.668	.668
HC (ANN)	.841	.841	.636	.636	.663	.663	.646	.646
HC (ANN) w/ disc.	.831	.831	.623	.623	.670	.670	.668	.668
MINCUT	-	-	-	-	-	-	-	-
MINCUT w/ disc.	-	-	-	-	-	-	-	-
MAJ	-	-	-	-	-	-	-	-
MAJ w/ disc.	-	-	-	-	-	-	-	-
LR (Imputation)	**.862**	.824	.705	.637	.750	.687	.675	.565
LR (Imputation) w/ disc.	.853	.764	.681	.607	.777	.643	.717	.665
ANN (Imputation)	**.862**	.824	**.706**	.638	**.781**	**.704**	.695	.565
ANN (Imputation) w/ disc.	.852	.764	.684	.608	.780	.642	**.718**	.664
RF (Imputation)	.828	.757	.648	.594	.763	.635	.703	.651
RF (Imputation) w/ disc.	.852	.761	.662	.594	.768	.628	**.718**	.659
kNN (Imputation)	.848	.771	.659	.580	.716	.554	.688	.623
kNN (Imputation) w/ disc.	.840	.735	.634	.561	.719	.540	.687	.596
LR (Reduced Feature)	.854	.778	.659	.603	.721	.622	.682	.638
LR (Reduced Feature) w/ disc.	.848	.753	.669	.617	.758	.640	.709	.673
ANN (Reduced Feature)	.849	.796	.655	.605	.728	.667	.685	.650
ANN (Reduced Feature) w/ disc.	.844	.792	.653	.607	.741	.666	.710	**.676**
RF (Reduced Feature)	.831	.756	.636	.592	.732	.548	.700	.650
RF (Reduced Feature) w/ disc.	.843	.778	.653	.617	.718	.564	.710	.671
kNN (Reduced Feature)	.828	.735	.626	.573	.695	.556	.685	.638
kNN (Reduced Feature) w/ disc.	.826	.735	.643	.592	.709	.561	.682	.635

Table 27. Our methods vs. the rest: mean classifier AUC for $\epsilon = 0.4$, unbalanced datasets ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.827	**.827**	.627	.627	.646	.646	.634	.634
HC (LR) w/ disc.	.820	.820	.637	**.637**	.661	.661	.660	.660
HC (ANN)	.808	.808	.613	.613	.659	.659	.637	.637
HC (ANN) w/ disc.	.796	.796	.607	.607	.642	.642	.659	.659
MINCUT	-	-	-	-	-	-	-	-
MINCUT w/ disc.	-	-	-	-	-	-	-	-
MAJ	-	-	-	-	-	-	-	-
MAJ w/ disc.	-	-	-	-	-	-	-	-
LR (Imputation)	**.842**	.794	.683	.620	.735	.674	.664	.560
LR (Imputation) w/ disc.	.827	.727	.663	.592	.754	.618	**.709**	.652
ANN (Imputation)	**.842**	.795	**.685**	.621	**.763**	**.687**	.678	.560
ANN (Imputation) w/ disc.	.826	.727	.664	.592	.756	.617	**.709**	.652
RF (Imputation)	.806	.736	.634	.584	.736	.628	.694	.649
RF (Imputation) w/ disc.	.826	.725	.645	.582	.744	.608	.708	.648
kNN (Imputation)	.823	.746	.638	.569	.701	.554	.677	.613
kNN (Imputation) w/ disc.	.812	.704	.617	.552	.696	.527	.671	.582
LR (Reduced Feature)	.830	.784	.633	.597	.704	.638	.669	.633
LR (Reduced Feature) w/ disc.	.824	.766	.653	.615	.736	.669	.696	.663
ANN (Reduced Feature)	.826	.786	.630	.596	.704	.659	.670	.641
ANN (Reduced Feature) w/ disc.	.820	.779	.637	.598	.710	.646	.696	**.664**
RF (Reduced Feature)	.807	.752	.618	.591	.701	.566	.686	.647
RF (Reduced Feature) w/ disc.	.823	.780	.641	.614	.703	.611	.696	.662
kNN (Reduced Feature)	.806	.747	.606	.574	.677	.577	.671	.629
kNN (Reduced Feature) w/ disc.	.806	.748	.628	.592	.687	.590	.668	.625

Table 28. Our methods vs. the rest: mean classifier AUC for $\epsilon = 0.5$, unbalanced datasets ("w/ disc." stands for "with discretization of features", "Tru." for "truthful reporting", and "Str." for "strategic reporting").

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC (LR)	.789	**.789**	.611	.611	.646	.646	.623	.623
HC (LR) w/ disc.	.779	.779	.616	**.616**	.655	.655	.646	**.646**
HC (ANN)	.776	.776	.597	.597	.643	.643	.625	.625
HC (ANN) w/ disc.	.767	.767	.584	.584	.624	.624	.645	.645
MINCUT	-	-	-	-	-	-	-	-
MINCUT w/ disc.	-	-	-	-	-	-	-	-
MAJ	-	-	-	-	-	-	-	-
MAJ w/ disc.	-	-	-	-	-	-	-	-
LR (Imputation)	**.815**	.749	.662	.601	.727	.665	.652	.555
LR (Imputation) w/ disc.	.791	.680	.640	.577	.727	.595	.695	.637
ANN (Imputation)	.814	.750	**.664**	.602	**.748**	**.676**	.660	.554
ANN (Imputation) w/ disc.	.790	.679	.640	.575	.728	.594	**.696**	.636
RF (Imputation)	.778	.698	.618	.572	.709	.619	.686	.644
RF (Imputation) w/ disc.	.791	.680	.627	.569	.719	.588	.693	.634
kNN (Imputation)	.790	.702	.616	.558	.684	.550	.666	.605
kNN (Imputation) w/ disc.	.774	.658	.595	.543	.672	.518	.658	.565
LR (Reduced Feature)	.792	.768	.609	.589	.682	.643	.653	.624
LR (Reduced Feature) w/ disc.	.785	.752	.633	.604	.706	.656	.673	.645
ANN (Reduced Feature)	.786	.760	.604	.583	.679	.641	.653	.629
ANN (Reduced Feature) w/ disc.	.782	.750	.611	.577	.672	.621	.673	.644
RF (Reduced Feature)	.771	.732	.599	.583	.667	.576	.667	.638
RF (Reduced Feature) w/ disc.	.788	.754	.624	.601	.682	.621	.673	.644
kNN (Reduced Feature)	.769	.740	.583	.574	.656	.588	.651	.620
kNN (Reduced Feature) w/ disc.	.770	.738	.608	.578	.658	.586	.648	.617

Supplementary References

Daniely, A., Sabato, S., Ben-David, S., and Shalev-Shwartz, S. Multiclass learnability and the erm principle. *The Journal of Machine Learning Research*, 16(1):2377–2404, 2015.

Dekel, O., Shamir, O., and Xiao, L. Learning to classify with missing and corrupted features. *Machine learning*, 81(2):149–178, 2010.

Dietterich, T. G. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7): 1895–1923, 1998.

Elkan, C. The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence*, volume 17, pp. 973–978. Lawrence Erlbaum Associates Ltd, 2001.

Globerson, A. and Roweis, S. Nightmare at test time: robust learning by feature deletion. In *Proceedings of the 23rd international conference on Machine learning*, pp. 353–360, 2006.

Lessmann, S., Baesens, B., Seow, H.-V., and Thomas, L. C. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, 247(1):124–136, 2015.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Shalev-Shwartz, S. and Ben-David, S. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Syed, U. and Taskar, B. Semi-supervised learning with adversarially missing label information. In *Advances in Neural Information Processing Systems*, pp. 2244–2252, 2010.

Vorobeychik, Y. and Kantarcioglu, M. Adversarial machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 12(3):1–169, 2018.