Interpretable Companions for Black-Box Models

Abstract

We present an interpretable *companion* model for any pre-trained black-box classifiers. The idea is that for any input, a user can decide to either receive a prediction from the black-box model, with high accuracy but no explanations, or employ a *companion rule* to obtain an interpretable prediction with slightly lower accuracy. The companion model is trained from data and the predictions of the black-box model, with the objective combining area under the transparency–accuracy curve and model complexity. Our model provides flexible choices for practitioners who face the dilemma of choosing between always using interpretable models and always using black-box models for a predictive task, so users can, for any given input, take a step back to resort to an interpretable prediction if they find the predictive performance satisfying, or stick to the black-box model if the rules are unsatisfying. To show the value of companion models, we design a human evaluation on more than seventy people to investigate the tolerable accuracy loss to gain interpretability for humans.

1 Introduction

The growing real-world needs for model understandability have triggered unprecedented advancement in the research in interpretable machine learning. Various forms of interpretable models have been created to compete with black-box models. Given a predictive task, users need to choose between a black-box model with high accuracy but no interpretability and an interpretable model with compromised performance. However, in many practices, the end-users, instead of the model designers, need to have the flexibility of choosing between whether they need an interpretable prediction with the slightly compromised predictive performance or a non-interpretable prediction but higher task performance, based on the input case they have.

We design a new mechanism of making a prediction in this paper. For any input, we provide users two options, to use a black-box model with high accuracy or to use a *companion* rule, which offers understandable prediction but with slightly lower accuracy. It is up to the user which type of prediction s/he prefers for any input. The rules are embedded in an "if-else" logic structure with decreasing accuracy. Therefore, if a user goes deeper into the list, more companion rules will be activated to cover more instances, gaining higher model transparency, but more considerable performance loss is incurred.

Our model is different from the current mainstream works in interpretable machine learning that focus on stand-alone interpretable models such as rule-based models [1] and linear models [2], or develops external black-box explanation methods [3, 4] to provide posthoc analysis. The former models may suffer from possible loss in accuracy since, aside from optimizing predictive performance, they also need to optimize model interpretability in parallel, which often conflicts with model fitness to the data. The latter type of methods, on the other hand, undergo heated debate on whether they consistently and truthfully reflect the underlying synergies between features inside a black-box [5].

Our model, *Companion Rule List* (CRL), takes a different route to avoid the possible weaknesses in the approaches above. Motivated by recent works on combining multiple models [6, 7], we pair a rule list with a pre-trained black-box model. Unlike previous hybrid models that create a fixed partition of the data space such that it is pre-determined which inputs will be processed by which model, we provide more flexibility to users by allowing users to switch between rules and the black-box, based on their task-specific and user-specific requirement for interpretability and predictive accuracy.

Figure 1 shows an example of CRL for evaluating customer credit. In CRL, we assume a black-box model with superior predictive performance is already obtained. CRL is designed to bring transparency into the decision making process as well as providing the

Preliminary work. Under review by AISTATS 2020. Do not distribute.

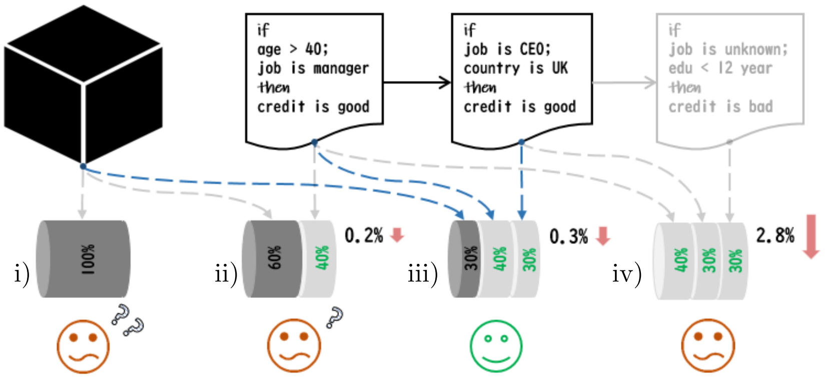


Figure 1: An example of evaluating customer credit with CRL. The users have four choices, i), ii), iii), and iv), for balancing the rule-based and the black-box predictions.

freedom of selecting the rule-based and the black-box predictions. In the figure, decision makers face four scenarios: i) using a completely black-box model; ii) adopting one rule that can explain 40% of the inputs but with losing accuracy for 0.2% and forwarding the rest 60% to the black-box; iii) adopting two rules that can explain 70% of the inputs but with losing accuracy for 0.3% and forwarding the rest 30% to the black-box; iv) adopting a completely transparent rule list but with losing accuracy for 2.8%. The final choice is made upon the users' preference on the trade-off between the transparency and the accuracy loss. For example, one may prefer adopting the scenario iii) as a compromising solution to the trade-off.

Note that, the goal of CRL is not to train a stand-alone accurate rule list, but to train a rule list with a productive collaboration with the black-box model such that they achieve an efficient trade-off between interpretability and predictive performance. To train CRL with a superior trade-off, we propose a novel training algorithm that can take the knowledge of the black-box model into account. Specifically, our algorithm is based on the area under the trade-off curve as the training objective function.

In the experiments, we demonstrate that the knowledge of the black-box partner during the training stage obtains a better performance for CRL, compared with other rule lists, such as CORELS [8] and SBRL [9] that are trained independently without knowledge of their black-box partner and paired with a black-box afterward.

To investigate whether users are willing to use companion rules instead of a black-box model despite possible performance loss, we conduct a carefully designed human evaluation on 79 participants with various background to study the conditions when a user would switch to a companion rule, specifically, how much accuracy a user is willing to give up in exchange for interpretability.

The rest of the paper is organized as follows. Section 2 reviews related work in machine learning model interpretation. Section 3 presents our proposed framework, CRL, for the collaboration of rule list and black-box. In Section 4, we propose a novel learning framework for CRL, and present a CRL training algorithm in Section 5. Section 6 provides an evaluation of CRL on data sets in several different domains, as well as a survey on how human subjects.

1.1 Preliminaries

Notation For any statement a, we denote the indicator function by $\mathbb{1}(a)$ where $\mathbb{1}(a) = 1$ if the statement a is true, and $\mathbb{1}(a) = 0$ otherwise.

Rule List Let $(x, y) \in \mathcal{X} \times \{0, 1\}$ be an observation consisting of an input feature vector x in a certain domain $\mathcal{X}$ and the output class label y. The rule list describes its prediction process in a "if-else-" format. Let the *decision function* d be a map from $\mathcal{X}$ to the Boolean domain, i.e. $d : \mathcal{X} \to \{\text{True}, \text{False}\}$. We refer to the pair of a decision function d and an output $z \in \{0, 1\}$ as a *rule* $r := (d, z)$. Here, we read the rule r as "if $d(x) = \text{True}$ for an input x, then the output $\hat{y} = z$". $d(x) = \text{True}$ means x satisfies the conditions in the rule. The rule list of length M consists of the sequence of M rules $R := (r_m = (d_m, z_m))_{m=1}^{M}$. The rule list returns the prediction $\hat{y}$ based on the following format.

$$\begin{aligned}
&\textbf{if } d_1(x) = \text{True}, \textbf{ then } \hat{y} = z_1 \\
&\textbf{else if } d_2(x) = \text{True}, \textbf{ then } \hat{y} = z_2 \\
&\dots \\
&\textbf{else if } d_M(x) = \text{True}, \textbf{ then } \hat{y} = z_M
\end{aligned}$$

Below we define the cover and prediction of rule lists.

Definition 1 (Cover)**.** We say that the m-th rule $r_m = (d_m, z_m)$ covers the input x if $d_m(x) = \text{True}$ and $d_k(x) = \text{False}$ for any of the previous rule $r_k = (d_k, z_k)$ with $k < m$. For the rule r_m and the input x, we define covers(r_m, x) by

$$\text{covers}(r_m, x) := \mathbb{1}\left(\left(\bigwedge_{k=1}^{m-1} \neg d_k(x)\right) \wedge d_m(x)\right). \quad (1)$$

Note that covers$(r_m, x) = 1$ if the input x is covered by r_m, and covers$(r_m, x) = 0$ otherwise. Thus, each input x is assigned to the first rule it satisfies.

Definition 2 (Rule List Prediction)**.** The prediction of the rule list of length M, consisting of the sequence of M rules $R := (r_m = (d_m, z_m))_{m=1}^{M}$, is defined by

$$\hat{y} = \mathcal{R}_M(x; R) := \sum_{m=1}^{M} z_m \text{covers}(r_m, x). \quad (2)$$

2 Related Work

In this section, we first elaborate distinctions from current interpretable machine learning researches and then discuss the similarity with techniques we inherit from existing works.

Distinction from Black-box Explainers CRL is not a black-box explainer. The rules are not designed to explain or approximate black-boxes like the rule-based explainers do [10, 11]. CRL is constructed to *compete with* and *locally replace* a black-box with competitive performance. We emphasize this distinction because black-box explainers serve the purpose of providing post-hoc analysis and approximations to the decision maker, which may not represent the true interactions of features inside the black-box [5, 12]. CRL, on the other hand, is the decision maker itself, and thus it truly represents how the decision is made.

Rule-Based Models Our CRL consists of decision rules. Rules are a well-adopted form of interpretable models for their language-like presentation and simple logic. Many state-of-the-art interpretable models are rule sets [1, 13, 14, 15] or rule lists [8, 9, 16] constructed via various learning algorithms like simulated annealing [17]. The idea is to propose a new model by making small changes to the current model, accept it with decreasing probability which is a function of how good the proposal is and the current temperature, until the maximum iterations are met or the solution converges. We design our training algorithm builds upon the prior wisdom from the works above and incorporate new strategies exploiting the unique structure of CRL.

Hybrid Models CRL is one type of companion models. Wang et al. [7] proposed to divide feature spaces into regions with sparse oblique trees and then assign black-box local experts to each region. Nan and Saligrama [18] designed a low-cost adaptive system by training a gating and prediction model that limits the utilization of a high-cost model to hard input instances and gates easy-to-handle input instances to a low-cost model. The work closest to ours is hybrid models [6, 19] that partition the feature space into transparency areas that are covered by rules and black-box area where rules fail to characterize.

Compared with our CRL, the models mentioned above create a fixed partition when the model is built. Thus it is determined at the training stage, by the model designer, which predictions will be processed by which model, and what level of transparency will be provided. CRL, on the other hand, leaves this decision to the end-user of the model, who may switch to different transparency and accuracy pairs for different inputs, accounting for various task-specific complications. Such flexibility is very critical and practical in domains where the final decision maker is human.

3 Companion Rule List

We propose *Companion Rule List* (CRL) as a collaboration framework for the interpretable and the black-box models. The advantage of our CRL lies in its flexibility that the users can choose which model to adopt for any inputs.

3.1 The Proposed Framework

Let f_b be a pre-obtained black-box model, such as the ensemble models or deep neural networks, that have superior predictive performance. CRL is a collaboration framework for the rule list and the black-box model. CRL is expressed in the following form.

if $d_1(x) = \text{True}$, **then** $\hat{y} = z_1$ **or** $\hat{y} = f_b(x)$
else if $d_2(x) = \text{True}$, **then** $\hat{y} = z_2$ **or** $\hat{y} = f_b(x)$
. . .
else if $d_M(x) = \text{True}$, **then** $\hat{y} = z_M$ **or** $\hat{y} = f_b(x)$

Note that CRL is flexible by its design. For any input x, if x is covered by the m-th rule as $d_m(x) = \text{True}$, the users have a choice of adopting the output $\hat{y} = z_m$ from the rule list or the output $\hat{y} = f_b(x)$ from the black-box model.

As we pointed out in Section 2, the hybrid rule set (HRS) [6, 19] also provides a framework for the collaboration of the interpretable and the black-box models. Despite the similarity, we would like to emphasize that HRS is not flexible. The rule set in the HRS covers a pre-determined fraction of the inputs. The end-users cannot choose between rules and the black-box model based on their own needs.

3.2 A Naive Implementation

A straightforward way to implement a companion model is to combine an independently trained rule list with the black-box model. For example, one can train a rule list by using CORELS [8] and SBRL [9]. Then, simply combining the trained rule list with the black-box model yields a companion model. This approach is straightforward, but the result can be suboptimal, as will be demonstrated by experiments. This is because the collaboration with the black-box model is not taken into account when training the rule list.

4 The Learning Framework for CRL

We discuss how we will construct a CRL to better collaborate with a black-box. To do that, we propose a novel objective function so that the interpretable and the black-box models collaborate more effectively than the naive implementation.

In what follows, we assume that an observation $(x, y) \in \mathcal{X} \times \{0, 1\}$ is sampled independently from an underlying distribution p. We also denote a set of independent observations (x, y) from p as the data set D, and denote its size by $|D|$.

4.1 Area Under the Transparency–Accuracy Trade-off Curve (AUTAC)

We propose *the area under the transparency–accuracy trade-off curve* (AUTAC) as a metric for general companion models.

Recall that we expect a companion model to be flexible so that the users can freely switch between the interpretable and the black-box models. Suppose that a user is willing to use the interpretable model for $100t\%$ of the observations (with $0 \leq t \leq 1$), and the black-box model for the remaining observations. Here, we refer to t as *transparency* of the companion model, which is a fraction of observations explained by the interpretable model. Note that a user preferring small t favors the black-box model because of its higher accuracy, while a user preferring large t favors the interpretable model. The difficulty here is that such a user's preference on t is unknown in practice. To bypass this difficulty, we adopt the following principle.

Maximum Accuracy Principle For any transparency t, a user prefers high accuracy.

From this principle, an ideal companion model is one that maximizes the accuracy for all transparency t. To this end, we propose the area under the transparency–accuracy trade-off curve (AUTAC) as the goodness measure of a companion model. Formally, let c_t be a companion model whose transparency is t, and let $A(c_t) := \mathbb{E}_{(x,y)\sim p}[\mathbb{1}(y = c_t(x)]$ be an accuracy of c_t. We can then draw a curve $(t, A(c_t))$ representing the transparency–accuracy trade-off. The area under the curve, or AUTAC, is then defined by

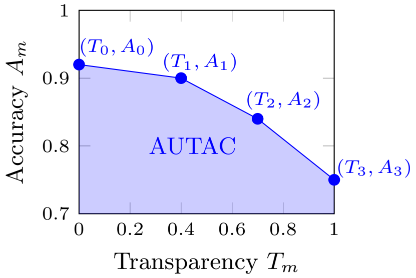


Figure 2: An example of the transparency–accuracy trade-off curve and the AUTAC of a Stochastic CRL with length $M = 3$.

$$\text{AUTAC} = \int_{t=0}^{1} A(c_t)dt. \tag{3}$$

4.2 AUTAC of CRL

We first define some important notions, and then derive AUTAC of CRL. Note that we generate rule lists with decreasing accuracy of rules. Thus, the maximum accuracy principle above implies that the users always adopt the first m consecutive rules in CRL as the interpretable model, where m can vary depending on their preferences on t.

Definition 3 (Transparency of Rule List)**.** For a rule list $\mathcal{R}_M(R)$ with a sequence of rules $R = (r_m = (d_m, z_m))_{m=1}^M$, the transparency of the first m rules is the probability of observations covered by these rules, defined by

$$T_m := \mathbb{E}_{(x,y)\sim p}\left[\mathbb{1}\left(\bigvee_{k=1}^{m} \text{covers}(r_k, x)\right)\right]. \tag{4}$$

Note that T_m takes a value between zero and one. The value one indicates that all observations can be explained by the first m rules, and thus its decision process is completely transparent. The value zero indicates that none of the observations are covered by the first m rules.

Definition 4 (Stochastic CRL)**.** Let $m^t := \max_{T_m \leq t} m$ be the maximum number of rules whose transparency is at most t. We define a Stochastic CRL with a transparency t by

if $\bigvee_{k=1}^{m^t} \text{covers}(r_k, x) = \text{True}$, **then** $y = \mathcal{R}_M(x; R)$
else if $\bigvee_{k=1}^{m^t+1} \text{covers}(r_k, x) = \text{False}$, **then** $y = f_b(x)$
else $y = \begin{cases} z_{m^t} & \text{if } q_t > \varepsilon \\ f_b(x) & \text{otherwise} \end{cases}$

where $q_t := \frac{t - T_{m^t}}{T_{m^t+1} - T_{m^t}}$, and ε is a uniform random variable in $[0, 1]$.

In Stochastic CRL, the users stochastically determine which model to adopt based on the transparency t and the random variable ε. Note that, the expected transparency of the Stochastic CRL with respect to ε can be easily verified as $q_t T_{m^t+1} + (1 - q_t) T_{m^t} = \frac{t - T_{m^t}}{T_{m^t+1} - T_{m^t}} T_{m^t+1} + \frac{T_{m^t+1} - t}{T_{m^t+1} - T_{m^t}} T_{m^t} = t$.

Defining AUTAC for CRL We define the AUTAC of CRL by adopting Stochastic CRL as the companion model c_t. Here, we use Stochastic CRL because its expected transparency t is continuous and the integral (3) is well-defined, while the transparency of the original CRL is defined only on discrete points $T_1, T_2, \ldots, T_M$. Let the accuracy of the Stochastic CRL be $A_m := \mathbb{E}_\varepsilon[A(c_{T_m})]$. We can then draw the transparency–accuracy trade-off curve as shown in Figure 2. Moreover, it is clear from the figure that AUTAC can be computed as follows.

Proposition 5 (AUTAC of Stochastic CRL)**.** The AUTAC (3) with Stochastic CRL as c_t is given by

$$\text{AUTAC}_M(R) = \frac{1}{2} \sum_{m=1}^{M} (A_m + A_{m-1})(T_m - T_{m-1}). \tag{5}$$

Here, we expressed the dependency of AUTAC to the sequence of the rules R and its length M, explicitly.

4.3 Learning Objective

We propose to train CRL so that the AUTAC is maximized. As a training objective, we use the following estimation of the AUTAC.

$$\widehat{\text{AUTAC}}_M(R) = \frac{1}{2} \sum_{m=1}^{M} (\hat{A}_m + \hat{A}_{m-1})(\hat{T}_m - \hat{T}_{m-1}),$$

where

$\hat{T}_m := \frac{|S_m|}{|D|}$,
$\hat{A}_m := \frac{1}{|D|} \sum_{(x,y) \in S_m} \mathbb{1}[y = \mathcal{R}_m(x; R)]$
$\qquad + \frac{1}{|D|} \sum_{(x,y) \in D \setminus S_m} \mathbb{1}[y = f_b(x)]$,
$S_m := \{(x, y) \mid \bigvee_{k=1}^{m} \text{covers}(r_k, x) = \text{True}, (x, y) \in D\}$.

The training of CRL is then formulated as

$$\max_M \max_R O_{M,\alpha}(R) := \widehat{\text{AUTAC}}_M(R) - \alpha M, \tag{6}$$

where $\alpha \geq 0$ is a parameter that penalizes the length of the rule list. Here, the additional term αM enforces the rule list to be sufficiently short so that it exhibits good interpretability.

5 Training Algorithm

We now design a training algorithm for CRL. Note that the problem (6) is a combinatorial optimization problem and finding a global optimum can take exponential time in general. We therefore take an alternative approach based on the heuristc search. Specifically, we adopt the stochastic local search for our training algorithm, which is shown to be effective for training high-quality rule models [15, 19].

The proposed training algorithm is shown in Algorithm 1. At the initilization step, we first apply FP-Growth [20] to the data set D, and find both the frequent positive rules and the frequent negative rules with the minimum support bounded by γ, and construct the set of rules Γ.[1] We then initialize the rule list with three rules chosen (possible randomly) from Γ. After the initialization is completed, we iteratively update the model using the stochastic local search. There are four possible operations for the model update: add, remove, swap, and replace.

- **Add**: We randomly select one rule from Γ. We then insert the selected rule into a random position in the current rule list.
- **Remove**: We randomly select one rule from the current rule list, and remove it from the list.
- **Switch**: We randomly select two rules in the current list, and swap their positions.
- **Replace**: We randomly select one rule from Γ. We also select one rule from the current rule list. We then replace these two rules.

In each update step, one of the four operations is randomly chosen and applied to the rule list. The model update is accepted with probability $\exp\left(\frac{O_{M,\alpha}(R^{[n+1]}) - O_{M,\alpha}(R^{[n]})}{C_0 / \log_2(1+n)}\right)$ which gradually decreases as the iteration n increases due to annealing.

6 Experiments

In the first part of this section, we test the performance of CRL on public datasets and compare it with

[1] Other rule miners such as Apriori or Eclat can also be used instead of FP-Growth.

Algorithm 1 Stochastic Local Search for CRL

```
Input: $f_b, D, \alpha, C_0$
▷Initialization
$\Gamma = ((d_j, z_j))_{j=1}^{J} \leftarrow$ FPGrowth$(D, \text{minsupp} = \gamma)$
  ▷ mine candidate rules from $D$
$R^{[0]} \leftarrow ((d_j, z_j))_{j=1}^{3}, R^* \leftarrow ((d_j, z_j))_{j=1}^{3}, M \leftarrow 3$
  ▷ initialize the rule list and the best solution

▷Stochastic Local Search
for $n = 1, 2, \dots, N$ do
    $\delta \sim$ random()
    if $\delta < \frac{1}{4}$ then   ▷Add
        $R^{[n+1]} \leftarrow$ add one rule to $R^{[n]}$ from $\Gamma$
        $M \leftarrow M + 1$
    else if $\delta < \frac{1}{2}$ then   ▷Remove
        $R^{[n+1]} \leftarrow$ remove one rule from $R^{[n]}$
        $M \leftarrow M - 1$
    else if $\delta < \frac{3}{4}$ then   ▷Swap
        $R^{[n+1]} \leftarrow$ swap two rules in $R^{[n]}$
    else   ▷Replace
        $R^{[n+1]} \leftarrow$ replace a rule in $R^{[n]}$ with a rule in $\Gamma$
    end if
    ▷Model Update
    $\epsilon \sim$ random()
    if $\epsilon > \exp\left(\frac{O_{M,\alpha}(R^{[n+1]}) - O_{M,\alpha}(R^{[n]})}{C_0 / \log_2(1+n)}\right)$ then
        $R^{[n+1]} \leftarrow R^{[n]}$
    end if
    $R^* \leftarrow \text{argmax}_{R \in \{R^{[n+1]}, R^*\}} O_{M,\alpha}(R)$
      ▷ update the best model
end for
output $\mathcal{R}_M(x; R^*)$   ▷Trained Rule List
```

two independently trained rule lists CORELS [8] and SBRL [9]. In the second part, a human evaluation of transparency–accuracy trade-off is conducted.

6.1 Experiments on Public Datasets

We evaluate the transparency–accuracy trade-off of CRL on eight public data sets from UCI machine learning repository [21] or ICPSR, listed in Table 1. Six data sets are associated with medical, financial, and judicial areas that have intensive demands for interpretable models. The other two data sets are associated with physics and commerce.

Preprocessing We preprocess the data by turning raw input into binary features so that rule mining is applicable. To do that, we convert numerical features to categorical features by using the quantile-based quantization.[2] We then apply one-hot encoding for categorical features (including the ones converted from the numerical features) to obtain binary features.

Table 1: Data Sets: N, d, and d' denote the number of instance, the number of features, and the number of binary features after preprocessing, respectively.

datasets	**category**	N	d	d'
messidor	medical	1151	19	202
german	finance	1000	20	160
adult	finance	48842	14	126
juvenile	justice	4023	55	315
frisk	justice	80755	26	92
recidivism	justice	11645	106	641
magic	physics	19020	10	140
coupon	commerce	3996	95	95

Baselines As discussed in section 3.2, we adopt the two rule list training algorithms, CORELS[3] [8] and SBRL[4] [9] as the baseline methods to be compared with. Note that, these algorithms does not utilize the information of their black-box partner when training the rule list. Thus, the obtained models from these training algorithms will become suboptimal, and will be outperformed by our proposed training algorithm. We also apply decision trees, C4.5 [22] and CART [23], to work as stand-alone interpretable baselines because they are one of the most popular interpretable models.

Setup To obtain a black-box f_b, we train RandomForest [24], AdaBoost [25], and XGBoost [26]. For the rule mining in Algorithm 1, we set the caridinality of each rule to be two and minimal support to be 0.05 so that rules capturing too few observations are eliminated. We also set the temperature $C_0 = 0.001$ and the training iteration $N = 50,000$. For all the models, we tuned their hyper-parameters so that the maximum number of rules to be less than 20.[5] All the models were trained and tested on a 5-fold cross validation. In each fold, we trained all the models using 80% of the data as train set, and evaluated their transparencies and the accuracies on the held out 20% test set.

Result We show the average transparency–accuracy trade-off curves for the eight data sets in Figure 3. Because the results are similar for all the black-box models, we selected RandomForest.[6] In the figures, for each of CRL, CORELS, and SBRL, we draw the average of the 5-fold as solid lines and the standard deviation as the shaded regions. The horizontal axis represents the transparency, and the vertical axis represents the accuracy. The average accuracies of CART

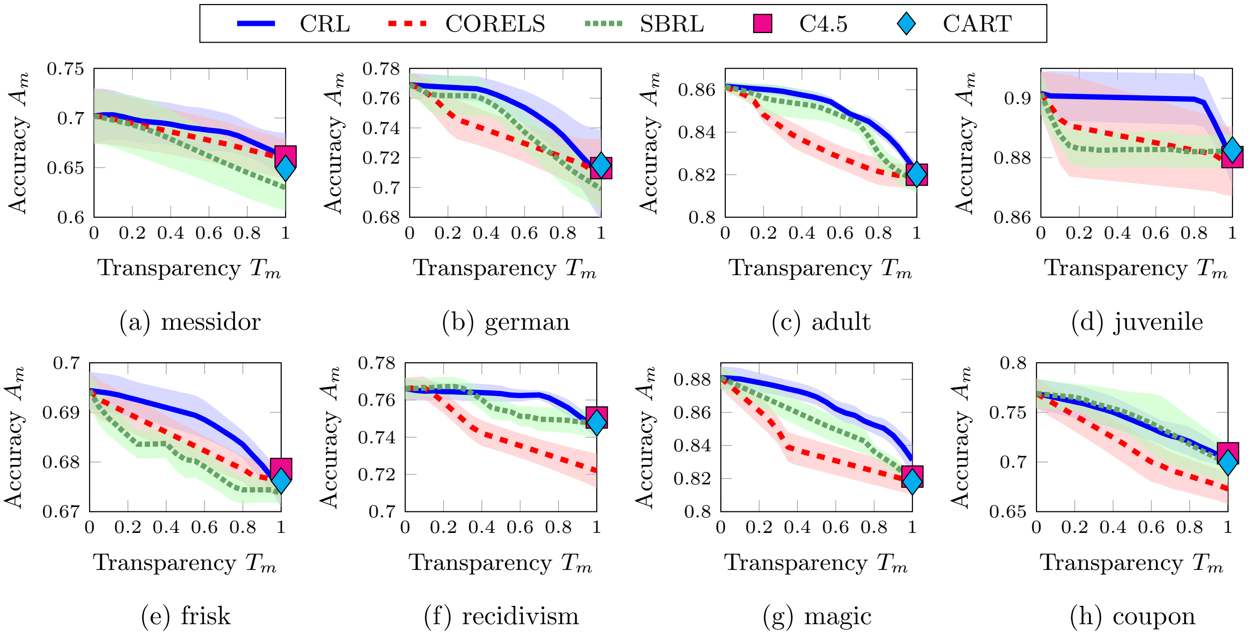
[2] We use `quantile_transform` in `scikit-learn` with the number of quantiles set to seven.

[3] `https://github.com/fingoldin/pycorels`

[4] `https://github.com/Hongyuy/sbrlmod`

[5] See Appendix A for the detailed parameter setups.

[6] See Appendix B for AdaBoost and XGBoost.



(a) messidor (b) german (c) adult (d) juvenile

(e) frisk (f) recidivism (g) magic (h) coupon

Figure 3: The transparency–accuracy trade-off (with RandomForest as f_b): The solid lines denote average trade-off curves, while the shaded regions denote ± standard deviations evaluated via 5-fold cross validation.

and C4.5 are shown as markers.

It is clearly observed that the curves of CRL are the highest in the figures for almost all data sets. This implies that the proposed CRL could provide the users with a flexible choice of the output with high accuracies. The lower curves of CORELS and SBRL indicate that these models are suboptimal because they did not collaborate with the black-box model f_b during training. It is also interesting to observe that CRL has comparable performance as other rule-based models such as CART and C4.5 when reaching transparency of 100% That is, the use of CRL as a stand-alone rule list is still a solid choice for the users. Thus, CRL alone satisfies users' need to go from transparency of 0% to 100%.

Table 2 summarizes AUTACs on all the data sets. The table shows the clear advantage of the proposed CRL that it successfully attained the better trade-off over CORELS and SBRL. This result confirms the effectiveness of the proposed training algorithm that can seek for a good collaboration of the rule list and the black-box.

An example In Table 3 we show an example of CRL model trained on the juvenile dataset, which predicts whether a child will commit delinquency, based on his prior growing-up experience. The data was collected via a survey where a child was asked to provide information about his prior exposure to violence from his friends, family, or community.

Table 2: AUTACs for all models. The numbers in the parenthesis denote standard deviations. Bold fonts denote the best results (underlined), and the results which was not significantly different from the best result (t-test with the 5% significance level).

	CRL	CORELS	SBRL
messidor	**.688 (.016)**	**.682 (.022)**	**.669 (.026)**
german	**.749 (.013)**	**.736 (.012)**	**.742 (.010)**
adult	**.851 (.001)**	.835 (.004)	**.847 (.005)**
juvenile	**.898 (.008)**	**.887 (.014)**	.884 (.005)
frisk	**.688 (.003)**	.684 (.003)	.681 (.003)
recidivism	**.761 (.004)**	.742 (.006)	.757 (.005)
magic	**.865 (.006)**	.842 (.006)	.854 (.007)
coupon	**.740 (.014)**	.716 (.017)	**.742 (.021)**

6.2 Human Evaluation of Transparency–Accuracy Trade-off

We evaluate the trade-off between transparency and accuracy for humans, specifically, how much is a person willing to sacrifice accuracy for transparency.

For this purpose, we designed a survey where we showed CRL models learned from three datasets, adult, german, and recidivism. For each prediction, we showed two options, a companion rule and a black-box model, and the estimated accuracy for both of them. Then we asked the participants to choose one from them. See the Appendix C for an example of the questions we designed. We designed 18 questions and

Table 3: An example of CRL on juvenile dataset (BLX represents black-box model)

	Companion Rule	$\hat{y}$	Rule cover	Rule acc.	BLX acc.
if	"Have your friends ever broken into a vehicle or building to steal something" = "No" **and** "Has anyone-including your family members or friends - ever attacked you with a gun, knife or some other weapon?"="Yes"	0	83.7%	93.3%	93.3%
else if	Sex = Male **and** "Have you ever tried cigarette smoking, even one or two puffs?" = "Yes"	1	6.6%	66.0%	69.7%
else if	"Have your friends ever used prescription drugs such as amphetamines or barbiturates when there was no medical need for them" = "No" **and** hard drug = 0	0	6.8%	61.8%	67.3%
else if	"Have your friends ever used alcohol" **and** Witness violence = TRUE	1	2.6%	52.4%	71.4%

randomly showed 12 of them for each participant.

We collected responses from 79 subjects in total and removed eight responses that fail the validation questions. The average age of the participants was 26.5, from the youngest 20 to the oldest 45. 60.3% of them were male. The majority of the participants were undergraduate and graduate students from the computer science department and business schools, who are current and future users of machine learning models. The rest were researchers from different domains, including business, medicine, and pharmacy, where interpretability is highly appreciated.

Figure 4 reports the percentages of participants who chose a companion rule over the black-box at different loss of accuracy of using a rule. The accuracy is reported in relative, i.e., the reduced accuracy for using a companion rule instead of the black-box divided by the accuracy of the black-box. The relative reduction in accuracy ranged from -25% to 5%, with the positive being rules that were better than the black-box model and negative being the ones that were worse. Results show that 80% of the participants were tolerable up to 4% accuracy loss or less. Increasing that loss to 5% lost about another 20% of the participants.

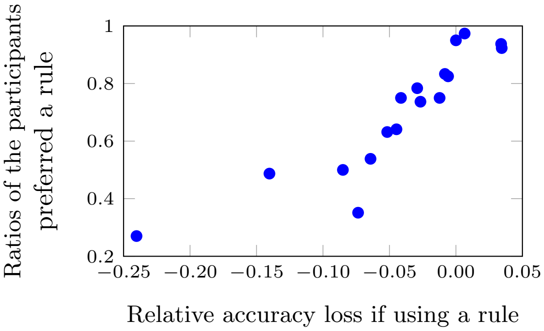


Figure 4: Human evaluated trade-off between model transparency and accuracy

Findings The results point out two interesting findings. First, while people desire interpretability, they are still very strict about accuracy. Interpretable models that lose too much accuracy are unlikely to be adopted in practice, which opens up opportunities for hybrid models like ours. Second, different users' preference for interpretability and accuracy can be so dramatically different (the curve covers a large range of accuracy drop while still having at least 30% users who prefer an interpretable model even at the loss of 25% of accuracy), thus it is important to leave the model selection at the end-user level, not the designer level. This is what CRL aims to do.

7 Conclusion

We proposed the companion rule list (CRL) for users to freely switch between the interpretable and the black-box models. CRL is flexible enough so that users can choose whether to adopt the rule-based prediction or the black-box prediction for any of the input, based on their preferences on the interpretability and the accuracy. We also designed a novel objective function, the area under the transparency–accuracy trade-off curve (AUTAC), for training a high-quality CRL, which we optimized by using the stochastic local search, utilizing information from the collaborative black-box model to form better collaboration between rules and the black-box model. Our result confirms that this collaborative training yields better companion rules than those rule lists trained independently.

Another main contribution of our paper is that we conducted an extensive human evaluation on 79 participants to study humans' views on model transparency versus accuracy. From the data we collected, we can understand how tolerable are humans to accuracy loss to gain model transparency. Results suggest that in practice, a stand-alone interpretable model will not be used by users if they drop too much in predictive performance. In addition, different people have a diverse preference for transparency, so the model selection should be made at the end-user level. Both open up opportunities for companion models.

References

[1] Tong Wang, Cynthia Rudin, Finale Doshi-Velez, Yimin Liu, Erica Klampfl, and Perry MacNeille. A bayesian framework for learning rule sets for interpretable classification. *The Journal of Machine Learning Research*, 18(1):2357–2393, 2017.

[2] Jiaming Zeng, Berk Ustun, and Cynthia Rudin. Interpretable classification models for recidivism prediction. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 180(3):689–722, 2017.

[3] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016.

[4] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017.

[5] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206, 2019.

[6] Tong Wang and Qihang Lin. Hybrid predictive model: When an interpretable model collaborates with a black-box model. *arXiv preprint arXiv:1905.04241*, 2019.

[7] Jialei Wang, Ryohei Fujimaki, and Yosuke Motohashi. Trading interpretability for accuracy: Oblique treed sparse additive models. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1245–1254. ACM, 2015.

[8] Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. Learning certifiably optimal rule lists. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 35–44. ACM, 2017.

[9] Hongyu Yang, Cynthia Rudin, and Margo Seltzer. Scalable bayesian rule lists. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3921–3930, 2017.

[10] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[11] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. Local rule-based explanations of black box decision systems. *arXiv preprint arXiv:1805.10820*, 2018.

[12] Ulrich Aïvodji, Hiromi Arai, Olivier Fortineau, Sébastien Gambs, Satoshi Hara, and Alain Tapp. Fairwashing: the risk of rationalization. In *Proceedings of the 36th International Conference on Machine Learning*, pages 161–170, 2019.

[13] Tong Wang. Multi-value rule sets for interpretable classification with feature-efficient representations. In *Advances in Neural Information Processing Systems*, pages 10835–10845, 2018.

[14] Sanjeeb Dash, Oktay Gunluk, and Dennis Wei. Boolean decision rules via column generation. In *Advances in Neural Information Processing Systems*, pages 4655–4665, 2018.

[15] Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1675–1684, 2016.

[16] Ulrich Aïvodji, Julien Ferry, Sébastien Gambs, Marie-José Huguet, and Mohamed Siala. Learning fair rule lists. *arXiv preprint arXiv:1909.03977*, 2019.

[17] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[18] Feng Nan and Venkatesh Saligrama. Adaptive classification for prediction under a budget. In *Advances in Neural Information Processing Systems*, pages 4727–4737, 2017.

[19] Tong Wang. Gaining free or low-cost interpretability with interpretable partial substitute. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6505–6514, 2019.

[20] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *ACM SIGMOD Record*, volume 29, pages 1–12. ACM, 2000.

[21] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[22] J Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[23] Leo Breiman. *Classification and regression trees*. Routledge, 2017.

[24] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[25] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[26] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

A Parameter Setups

We set the parameters of the proposed training algorithm, CORELS, and SBRL, as follows.

Proposed Algorithm We tuned the penalty parameter α in the training objective function (6) based on Algorithm 2. For all the data sets, we mined rules containing maximally two conditions. When the 16 GB memory on our machine was found to be insufficient for rule mining, we mined rules on a subset of the data set by reducing the number of observations. We searched for the optimal α from the candidates ranging from 10^{-4} to 10^{-2}. We determined the optimal α as the one that maximized the AUTAC on the training set D, under the constraint that number of conditions is less than 20.

Algorithm 2 CRL Tuning Strategy

```
//Initial setting
$I_{\max} \leftarrow 20$   ▷max rules
$C_{\max} \leftarrow 2$   ▷max conditions
$A \leftarrow [.01, .005, .001, .0008, .0005, .0002, .0001]$
  ▷candidates of $\alpha$
$\alpha_{\text{opt}} \leftarrow 0$, $\text{AUTAC}_{\text{opt}} \leftarrow 0$
FP-Growth minimal support $\leftarrow 0.05$

//Tuning rule mining
$b \leftarrow |D|$   ▷# of observations for mining
while memory insufficient do
  $b \leftarrow \lfloor 0.9b \rfloor$
  mine rules on $b$ observations with FP-Growth
end while

//Tuning $\alpha$
for $\alpha \in A$ do
  $M, \text{AUTAC} \leftarrow \text{train}(D, \alpha, C_{\max})$
  if $M < I_{\max}$ then
    if $\text{AUTAC} > \text{AUTAC}_{\text{opt}}$ then
      $\text{AUTAC}_{\text{opt}} \leftarrow \text{AUTAC}$
      $\alpha_{\text{opt}} \leftarrow \alpha$
    end if
  end if
end for
Output $\alpha_{\text{opt}}$
```

CORELS For CORELS, we used an implementation publicly available at `https://github.com/fingoldin/pycorels`. We tuned the maximal number of iterations N and policy P based on Algorithm 3. We increment N by $50k$ each time until $30k$ and search the best policy P in list L. For all the data sets, we mined rules containing maximally two conditions. We determined the optimal N and P when predictive accuracy ACC is maximized on the testing set, under the constraint that condition number is less than 20.

Algorithm 3 CORELS Tuning Strategy

```
//Initial setting
$I_{\max} \leftarrow 20$   ▷max rules
$C_{\max} \leftarrow 2$   ▷max conditions
$N \leftarrow 100\text{k}$   ▷maximum number of iterations
$L \leftarrow [curious, lowerbound, dfs, bfs, objective]$
  ▷candidates of $P$
$N_{\text{opt}} \leftarrow 0$, $\text{ACC}_{\text{opt}} \leftarrow 0$, $P_{\text{opt}} \leftarrow \text{None}$

//Tuning $N$ and $P$
for $t = 1, 2...5$ do
  for $P \in L$ do
    $M, \text{ACC} \leftarrow \text{train}(D, N, P, C_{\max})$
    if $M < I_{max}$ then
      if $\text{ACC} > \text{ACC}_{\text{opt}}$ then
        $\text{ACC}_{\text{opt}} \leftarrow \text{ACC}$
        $N_{\text{opt}} \leftarrow N$
        $P_{\text{opt}} \leftarrow P$
      end if
    end if
  end for
  $N \leftarrow N + 50k$
end for
output $N_{\text{opt}}$, $P_{\text{opt}}$
```

SBRL For SBRL, we used an implementation publicly available at `https://github.com/Hongyuy/sbrlmod`. We tuned number of chains Nc and the expected length of the rule list λ based on Algorithm 4. We mine rules maximally containing two rules on messidor, german, adult, magic and coupon. When the 16 GB memory on our machine was found to be insufficient for rule mining, we increment both positive and negative minimal support S_+ and S_- by 0.05. When the minimal support is higher than an threshold T, we turn to mine rules maximally containing one rules. We mine rules maximally containing one rules on juvenile, frisk and recidivism. We determined the optimal λ and Nc as the ones that maximized predictive accuracy ACC on the testing set, under the constraint that number of conditions is less than 20.

Algorithm 4 SBRL Tuning Strategy

```
//Initial setting
$I_{\max} \leftarrow 20$   ▷max rules
$C_{\max} \leftarrow 2$   ▷max conditions
$T \leftarrow 0.7$   ▷threshold to reduce $C_{max}$
$\Lambda \leftarrow [1, 2, 5, 10, 15, 20, 25, 30]$
  ▷candidates of $\lambda$
$Nc \leftarrow 0$  ▷number of chains
$\lambda_{\text{opt}} \leftarrow 0$, $Nc_{\text{opt}} \leftarrow 0$, $\text{ACC}_{\text{opt}} \leftarrow 0$
$S_+ \leftarrow 0.05$, $S_- \leftarrow 0.05$
 ▷minimal support for pos and neg rules

//Tuning minimal support
while memory insufficient do
  $S_+ \leftarrow S_+ + 0.05$
  $S_- \leftarrow S_- + 0.05$
  if $S_+ > T$ then
    $C_{\max} \leftarrow 1$
  end if
end while

//Tuning $\lambda$ and $Nc$
for $\lambda \in \Lambda$ do
  for $t = 1...6$ do
    $M, \text{ACC} \leftarrow \text{train}(D, \lambda, Nc, S_+, S_-, C_{max})$
    $Nc \leftarrow Nc + 5$
    if $M < I_{max}$ then
      if $\text{ACC} > \text{ACC}_{\text{opt}}$ then
        $\text{ACC}_{\text{opt}} \leftarrow \text{ACC}$
        $\lambda_{\text{opt}} \leftarrow \lambda$
        $Nc_{\text{opt}} \leftarrow Nc$
      end if
    end if
  end for
end for
output $\lambda_{\text{opt}}$, $Nc_{\text{opt}}$
```

B Exhaustive Results

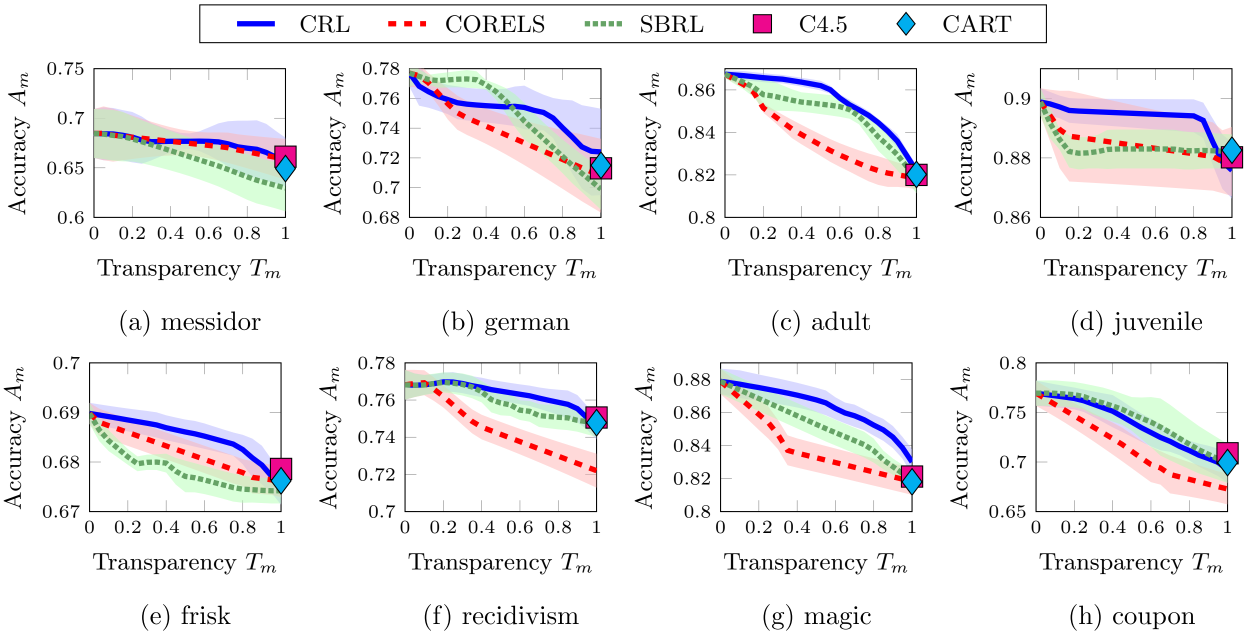
Here, we show the results for AdaBoost and XGBoost we omitted in Section 6 due to space limitation. Figures 5 and 6 show the trade-off curves, and Tables 4 and 5 show AUTACs. These results also confirm the validity of the proposed training algorithm.

Table 4: AUTACs on AdaBoost: The numbers in the parenthesis denote standard deviations. Bold fonts denote the best results (underlined), and the results which was not significantly different from the best result (t-test with the 5% significance level).

	CRL	CORELS	SBRL
messidor	**.675 (.010)**	**.674 (.013)**	**.660 (.020)**
german	**.754 (.010)**	.738 (.013)	**.749 (.005)**
adult	**.856 (.002)**	.837 (.004)	.850 (.004)
juvenile	**.892 (.005)**	**.885 (.013)**	.883 (.005)
frisk	**.685 (.003)**	.682 (.003)	.678 (.002)
recidivism	**.763 (.006)**	.744 (.007)	.759 (.006)
magic	**.864 (.007)**	.841 (.010)	.852 (.007)
coupon	**.740 (.012)**	.714 (.015)	**.743 (.017)**

Table 5: AUTACs on XGBoost: The numbers in the parenthesis denote standard deviations. Bold fonts denote the best results (underlined), and the results which was not significantly different from the best result (t-test with the 5% significance level).

	CRL	CORELS	SBRL
messidor	**.689 (.017)**	**.681 (.019)**	**.670 (.022)**
german	**.748 (.022)**	**.732 (.016)**	**.734 (.010)**
adult	**.856 (.002)**	.837 (.004)	.851 (.004)
juvenile	**.898 (.006)**	**.888 (.015)**	.884 (.005)
frisk	**.686 (.003)**	.683 (.003)	.679 (.003)
recidivism	**.766 (.007)**	.744 (.008)	.759 (.006)
magic	**.863 (.004)**	.840 (.007)	.853 (.004)
coupon	**.739 (.013)**	.713 (.015)	**.744 (.017)**



(a) messidor

(b) german

(c) adult

(d) juvenile

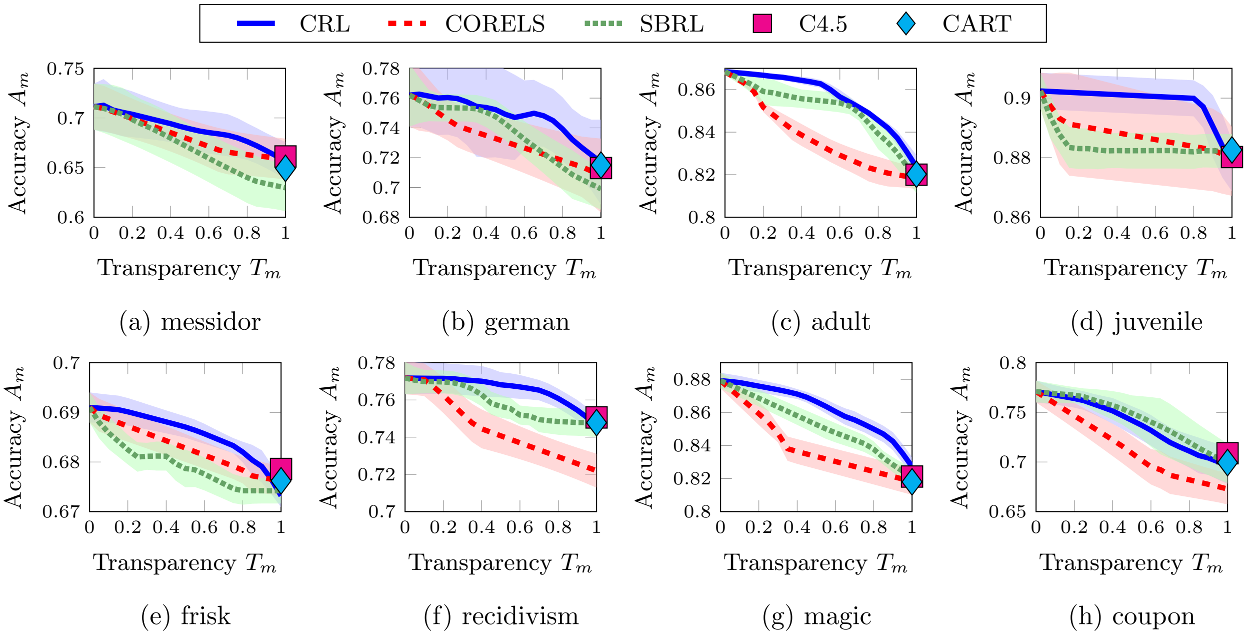
(e) frisk

(f) recidivism

(g) magic

(h) coupon

Figure 5: The transparency–accuracy trade-off (with AdaBoost as f_b): The solid lines denote average trade-off curves, while the shaded regions denote ± standard deviations evaluated via 5-fold cross validation.



(a) messidor

(b) german

(c) adult

(d) juvenile

(e) frisk

(f) recidivism

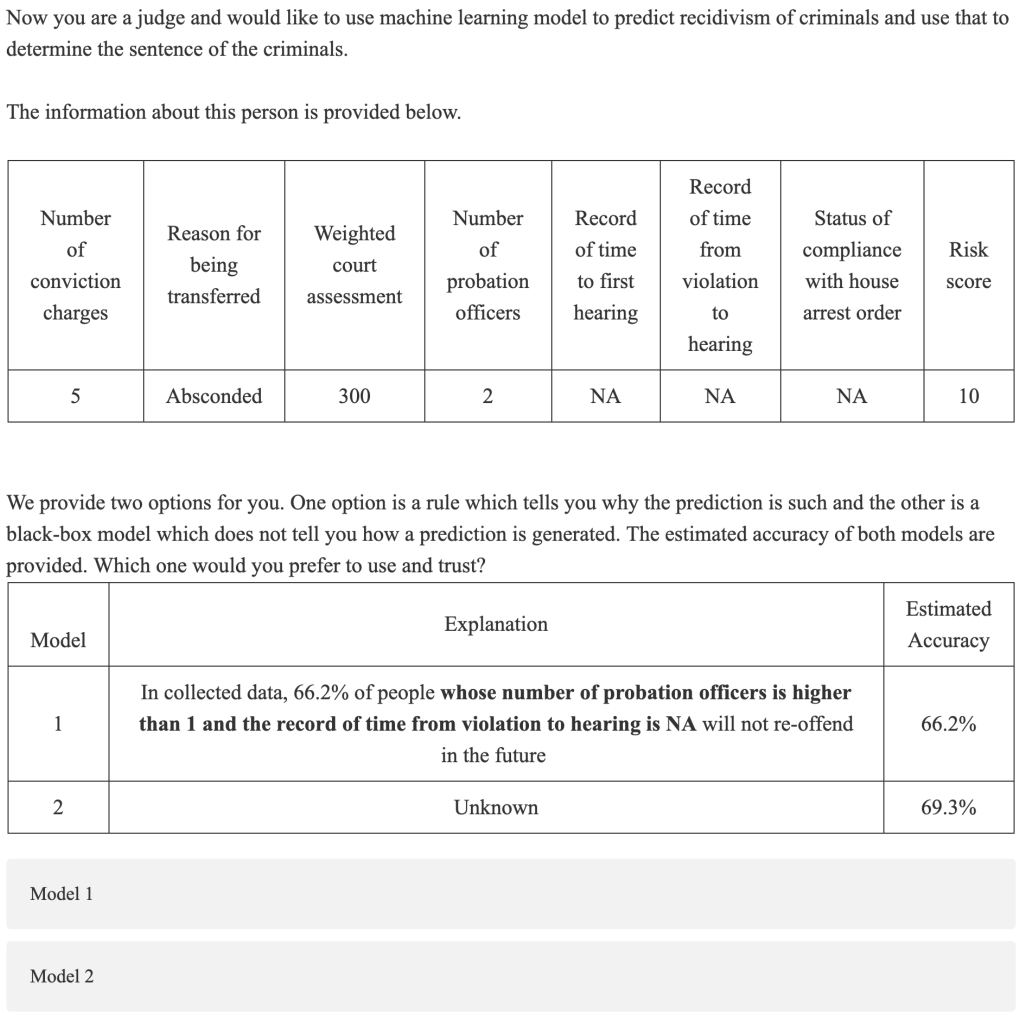
(g) magic

(h) coupon

Figure 6: The transparency–accuracy trade-off (with XGBoost as f_b): The solid lines denote average trade-off curves, while the shaded regions denote ± standard deviations evaluated via 5-fold cross validation.

C Survey

Figure 7 is an example of our survey question.



Now you are a judge and would like to use machine learning model to predict recidivism of criminals and use that to determine the sentence of the criminals.

The information about this person is provided below.

Number of conviction charges	Reason for being transferred	Weighted court assessment	Number of probation officers	Record of time to first hearing	Record of time from violation to hearing	Status of compliance with house arrest order	Risk score
5	Absconded	300	2	NA	NA	NA	10

We provide two options for you. One option is a rule which tells you why the prediction is such and the other is a black-box model which does not tell you how a prediction is generated. The estimated accuracy of both models are provided. Which one would you prefer to use and trust?

Model	Explanation	Estimated Accuracy
1	In collected data, 66.2% of people **whose number of probation officers is higher than 1 and the record of time from violation to hearing is NA** will not re-offend in the future	66.2%
2	Unknown	69.3%

Figure 7: An example of our survey questions.