Interpretable Differencing of Machine Learning Models

Swagatam Haldar[1] **Diptikalyan Saha**[1] **Dennis Wei**[2] **Rahul Nair**[3] **Elizabeth M. Daly**[3]

[1]IBM Research, Bangalore, India
[2]IBM Research, Yorktown Heights, New York, USA,
[3]IBM Research, Dublin, Ireland

Abstract

Understanding the differences between machine learning (ML) models is of interest in scenarios ranging from choosing amongst a set of competing models, to updating a deployed model with new training data. In these cases, we wish to go beyond differences in overall metrics such as accuracy to identify *where* in the feature space do the differences occur. We formalize this problem of model *differencing* as one of predicting a dissimilarity function of two ML models' outputs, subject to the representation of the differences being human-interpretable. Our solution is to learn a *Joint Surrogate Tree* (JST), which is composed of two conjoined decision tree surrogates for the two models. A JST provides an intuitive representation of differences and places the changes in the context of the models' decision logic. Context is important as it helps users to map differences to an underlying mental model of an AI system. We also propose a refinement procedure to increase the precision of a JST. We demonstrate, through an empirical evaluation, that such contextual differencing is concise and can be achieved with no loss in fidelity over naive approaches.

1 INTRODUCTION

At various stages of the AI model lifecycle, data scientists make decisions regarding which model to use. For instance, they may choose from a range of pre-built models, select from a list of candidate models generated from automated tools like AutoML, or simply update a model based on new training data to incorporate distributional changes. In these settings, the choice of a model is preceded by an evaluation that typically focuses on accuracy and other metrics, instead of how it differs from other models.

We address the problem of model *differencing*. Given two models for the same task and a dataset, we seek to learn where in the feature space the models' predicted outcomes differ. Our objective is to provide accurate and interpretable mechanisms to uncover these differences.

The comparison is helpful in several scenarios. In a *model marketplace*, multiple pre-built models for the same task need to be compared. The models usually are black-box and possibly trained on different sets of data drawn from the same distribution. During *model selection*, a data scientist trains multiple models and needs to select one model for deployment. In this setting, the models are white-box and typically trained on the same training data. For *model change*, where a model is retrained with updated training data with a goal towards model improvement, the data scientist needs to understand changes in the model beyond accuracy metrics. Finally, *decision pipelines consisting of logic and ML models* occur in business contexts where a combination of business logic and the output of ML models work together for a final output. Changes might occur either due to model retraining or adjustments in business logic which can impact the behavior of the overall pipeline.

In this work we address the problem of interpretable model differencing as follows. First, we formulate the problem as one of predicting the values of a dissimilarity function of the two models' outputs. We focus herein on 0-1 dissimilarity for two classifiers, where 0 means "same output" and 1 means "different", so that prediction quality can be quantified by any binary classification metric such as precision and recall. Second, we propose a method that learns a *Joint Surrogate Tree* (JST), composed of two conjoined decision tree surrogates to jointly approximate the two models. The root and lower branches of the conjoined decision trees are common to both models, while higher branches (farther from root) may be specific to one model. A JST thus accomplishes two tasks at once: it provides interpretable surrogates for the two models while also aligning the surrogates for easier comparison and identification of differences. These aspects are encapsulated in a visualization of JSTs that we

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

present. Third, a refinement procedure is used to grow the surrogates in selected regions, improving the precision of the dissimilarity prediction.

Our design of jointly learning surrogates is motivated by the need to place model differences in the context of the overall decision logic. This can aid users who may already have a mental model of (individual) AI systems, either for debugging [Kulesza et al., 2012] or to understand errors [Bansal et al., 2019].

The main contributions of the paper are (a) a quantitative formulation of the problem of model differencing, and (b) algorithms to learn and refine conjoined decision tree surrogates to approximate two models simultaneously. A detailed evaluation of the method is presented on several benchmark datasets, showing more accurate or more concise representation of model differences, compared to baselines.

2 RELATED WORKS

Our work touches upon several active areas of research which we summarize based on key pertinent themes.

Surrogate models and model refinement One mechanism to lend interpretability to machine learning models is through surrogates, i.e., simpler human-readable models that mimic a complex model [Buciluǎ et al., 2006, Ba and Caruana, 2014, Hinton et al., 2015, Lopez-Paz et al., 2016]. Most relevant to this paper are works that use a decision tree as the surrogate [Craven, 1996, Bastani et al., 2017, Frosst and Hinton, 2017]. Bastani et al. [2017] showed that interpretable surrogate decision trees extracted from a black-box ML model allowed users to predict the same outcome as the original ML model. Freitas [2014] also discusses interpretability and usefulness of using decision trees as surrogates. None of these works however have considered jointly approximating two black-box models.

Decision tree generation with additional objectives Chen et al. [2019] showed that decision tree generation is not robust and slight changes in the root node can result in a very different tree structure. Chen et al. [2019], Andriushchenko and Hein [2019] focus on improving robustness when generating the decision tree while Moshkovitz et al. [2021] prioritises both robustness and interpretability. Aghaei et al. [2019] use mixed-integer optimization to take fairness into account in the decision tree generation. However, none of these solutions consider the task of comparing two decision trees.

Predicting disagreement or shift Prior work has focused on identifying statistically whether models have significantly changed [Bu et al., 2019, Geng et al., 2019, Harel et al., 2014], but not on where they have changed. Cito et al. [2021] present a model-agnostic rule-induction algorithm to produce interpretable rules capturing instances that are mispredicted with respect to their ground truth.

Comparing models The "distill-and-compare" approach of Tan et al. [2018] uses generalized additive models (GAMs) and fits one GAM to a black-box model and a second GAM to ground truth outcomes. While differences between the GAMs are studied to uncover insights, there is only one black-box model. Demšar and Bosnić [2018] study concept drift by determining feature contributions to a model and observing the changes in contributions over time. Similarly, Duckworth et al. [2021] investigated changes in feature importance rankings pre- and post-COVID. This approach however does not localize changes to regions of the feature space. Chouldechova and G'Sell [2017] compare models in terms of fairness metrics and identify groups in the data where two models have maximum disparity. Prior work by Nair et al. [2021], which is most similar to our own, uses rule-based surrogates for two models and derives rules for where the models behave similarly. Their method biases the learning of the second surrogate based on inputs from the first model, a step they call grounding, and imposes a one-to-one mapping between rules in the two surrogates. This is a strict condition that may not hold in practice. Additionally, their method does not evaluate the accuracy of resulting rules in predicting model similarities or differences. Our approach addresses these limitations.

3 PROBLEM STATEMENT AND PRELIMINARIES

We are given two predictive models $M_1, M_2 : \mathcal{X} \to \mathcal{Y}$ mapping a feature space $\mathcal{X} \subset \mathbb{R}^d$ to an output space $\mathcal{Y}$, as well as a *dissimilarity function* $D : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ (where $\mathbb{R}_+$ means the non-negative reals including zero) for comparing the outputs of the two models. Our goal is to obtain a *difference model* ("diff-model" for short), $\hat{D} : \mathcal{X} \to \mathbb{R}_+$, that predicts the dissimilarity $D(M_1(x), M_2(x))$ well while also being interpretable. To construct $\hat{D}$, we assume access to a dataset $X \in \mathbb{R}^{n \times d}$ consisting of n samples drawn i.i.d. from a probability distribution P over $\mathcal{X}$. This dataset does not have to have ground truth labels, in contrast to supervised learning, since supervision is provided by the models M_1, M_2. Prediction quality is measured by the expectation $\mathbb{E}[L(\hat{D}(X), D(M_1(X), M_2(X))]$ of one or more metrics $L : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ comparing $\hat{D}$ to D, where the expectation is with respect to P. In practice, these expectations are approximated empirically using a test set.

In this work, we focus on classification models M_1 and M_2, so that $\mathcal{Y}$ is a finite set, and 0-1 dissimilarity $D(M_1(x), M_2(x)) = 1$ if $M_1(x) \neq M_2(x)$ and $D(M_1(x), M_2(x)) = 0$ otherwise. Accordingly, the predictions $\hat{D}(x)$ are also binary-valued and any binary classification metrics L may be used for evaluation. Herein we use precision, recall, and F1-score (described in Section 5).

We use decision trees as the basis for our Joint Surrogate Tree solution. To ensure interpretability, the height (also referred to as maximum depth) is constrained to a small value (e.g. 6 in our experiments). Below we define notation and terminology related to decision trees for later use.

Decision Tree A decision tree is a binary tree $T = (V_{dt}, E_{dt})$ with a node set V_{dt}, a root node $r \in V_{dt}$ and a directed set of edges $E_{dt} \subset V_{dt} \times V_{dt}$. Each internal node $v \in V_{dt}$ contains a split condition $s(v) := f(v) < t(v)$ containing a predicate on feature $f(v) \in [d]$ (where $[d]$ is the shorthand for $\{1 \dots d\}$), and a threshold $t(v) \in \mathbb{R}$, and two children v_T and v_F. The edges (v, v_T) and (v, v_F) are annotated with edge conditions $f(v) < t(v)$ and $f(v) \geq t(v)$, respectively. Each leaf node v contains a label $\text{label}(v) \in \mathcal{Y}$. All leaf nodes of a tree rooted at r are denoted as $\text{leaves}(r)$. Given a node v, path-condition of v (denoted as $\text{pc}(v)$) is defined as the conjunction of all edge conditions from r to v. At a given node $v \in V_{dt}$, we denote by X_v, y_v the subset of samples that satisfy the $pc(v)$ and their labels, and we use $X_v[f]$ to denote the set of values for the feature $f \in [d]$. Without loss of generality, $s(v)$ is formed by minimizing function H, for all features and their values. We express the split condition at node v as $s(v) = c(X_v, y_v)$ and the minimum objective value (impurity) by $\text{imp}(X_v, y_v)$:

$$c(X_v, y_v) = \underset{\{f \in [d],\, t \in X_v[f]\}}{\arg\min} H(f, t, X_v, y_v) \quad (1)$$

$$\text{imp}(X_v, y_v) = \min_{\{f \in [d],\, t \in X_v[f]\}} H(f, t, X_v, y_v) \quad (2)$$

For example, H can be instantiated as the weighted sum of entropy values of left and right split [Quinlan, 1986]. We now describe two baseline approaches to the problem before presenting our proposed algorithm in Section 4.

Direct difference modelling Given the above problem statement, a natural way to predict the dissimilarity function D is to let $\hat{D}$ be a single ML model, in our case a decision tree for interpretability, and train it to classify between $D = 0$ (models M_1, M_2 having the same output) and $D = 1$ (different output). We call this the *direct* approach. The main drawback of direct differencing is that even when using an interpretable decision tree, it does not capture the differences between the two models in the context of their human-interpretable decision processes, i.e., where in the decision logic of the models do the differences occur.

Surrogate modelling Another natural way to model the dissimilarity is to separately build a decision tree surrogate $\hat{M}_i$ for each input model M_i, $i = 1, 2$, using the outputs of M_i on the input samples X for training the surrogate. Then we predict $\hat{D}(x) = 1$ if $\hat{M}_1(x) \neq \hat{M}_2(x)$ and $\hat{D}(x) = 0$ otherwise. We call this the *separate surrogate* approach. Its drawback is that the two decision tree surrogates are not aligned, making it cumbersome for human comparison. In Section 5, we show that the manifestation of this drawback is the large number of rules (see next paragraph) needed to describe all the regions where the two surrogates differ.

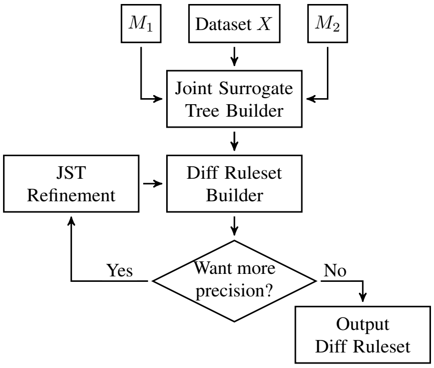

Figure 1: Method Overview

Diff rules as output We use *diff rules* as an interpretable representation of model differences for both direct and surrogate tree-based diff models. A diff rule is a conjunction of conditions on individual features that, when satisfied at a point x, yields the prediction $\hat{D}(x) = 1$. Corresponding to each diff rule is a *diff region*, the set of x's that satisfy the rule. A *diff ruleset* $\mathcal{R}$ is a set of diff rules such that if x satisfies any rule in the set, we predict $\hat{D}(x) = 1$. For a direct decision tree model, the diff rules are given by the path conditions of the $\hat{D}(x) = 1$ leaves. For surrogate models $\hat{M}_1, \hat{M}_2$, the diff rules are conjunctions of path conditions for pairs of intersecting leaves where $\hat{M}_1(x) \neq \hat{M}_2(x)$.

4 PROPOSED ALGORITHM

We propose a technique called IMD, which shows the differences between two ML models by constructing a novel representation called a *Joint Surrogate Tree* or JST. A JST is composed of two conjoined decision tree surrogates that jointly approximate the two models, intuitively capturing similarities and differences between them. It overcomes the drawbacks of the direct and separate surrogate approaches mentioned in Section 3: it avoids the non-smoothness of direct difference modelling, aligns and promotes similarity between surrogates for the two models, and shows differences in the context of each model's decision logic. Our method has a single hyperparameter, tree depth, which controls the trade-off between accuracy and interpretability.

IMD performs two steps as shown in Figure 1. In the first step, IMD builds a JST for models M_1, M_2 using data samples X, and then extracts diff regions from the JST. Interpretability is ensured by restricting the height of the JST. The IMD algorithm treats M_1, M_2 as black boxes and can handle any pair of classification models. It is also easy to implement as it requires a simple modification to popular

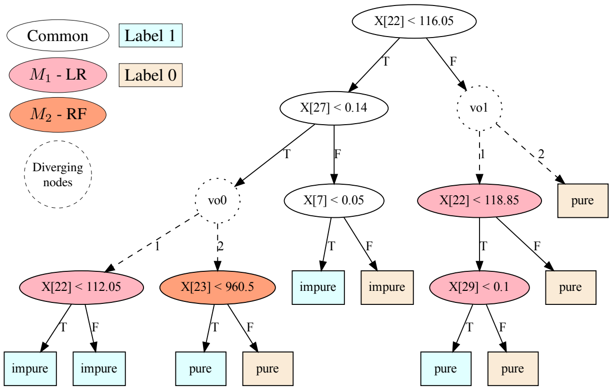

Figure 2: A JST for the Breast Cancer (*bc*) dataset.

greedy decision tree algorithms.

The second (optional) step, discussed at the end of Section 4.1, refines the JST by identifying diff regions where the two decision tree surrogates within the JST differ but the original models do not agree with the surrogates on their predictions. The refinement process aims to increase the fidelity of the surrogates in the diff regions, thereby generating more precise diff regions where the true models also differ.

4.1 JOINT SURROGATE TREE

Representation Figure 2 shows an example of a JST for Logistic Regression and Random Forest models on the Breast cancer dataset [Dua and Graff, 2017] (feature names are omitted to save space). The JST consists of two conjoined decision tree surrogates for the two models. The white oval nodes of the JST are shared decision nodes where both surrogates use the same split conditions. We refer to the subtree consisting of white nodes as the common prefix tree. In contrast, the colored nodes represent separate decision nodes, pink for surrogate $\hat{M}_1$ corresponding to M_1, and orange for surrogate $\hat{M}_2$ for M_2. The rectangular nodes correspond to the leaves, and are colored differently to represent class labels — cyan for label 1, and beige for label 0. The leaves are marked as pure/impure depending on whether all the samples falling there have the same label or not.

The JST intuitively captures diff regions, i.e., local regions of feature space where the two input models diverge, and also groups them into a two-level hierarchy. As with any surrogate-based diff model, we have $\hat{D}(x) = 1$ if and only if the constituent decision tree surrogates disagree, $\hat{M}_1(x) \neq \hat{M}_2(x)$. Thus, diff regions can be identified by first focusing on an *or-node* (the dotted circle nodes in Figure 2 where the surrogates diverge) and then enumerating pairs of leaves under it with different labels.

For example, considering the rightmost or-node v_{o1} in Figure 2, with path condition $X[22] \geq 116.05$, $\hat{M}_2$ classifies all the samples to label 0 whereas $\hat{M}_1$ classifies to label 1 in the region $X[22] < 118.85 \wedge X[29] < 0.1$. Therefore the diff region is $118.85 > X[22] \geq 116.05 \wedge X[29] < 0.1$. While in this case v_{o1} yields a single diff region, in general multiple diff regions could be grouped under a single or-node, resulting in a hierarchy. By processing all the or-nodes of the JST, one obtains all diff-regions.

Formally, $JST = (V = V_{dt} \cup V_o, E = E_{dt} \cup E_o)$. V_{dt} is a set of decision nodes similar to decision trees (oval shaped in figure) with each outgoing edge $\in E_{dt}$ (solid arrows) representing True or False decisions as in a regular decision tree. V_o are the set of or-nodes (circular nodes) representing the diverging points where the decision trees no longer share the same split conditions. Each child of $v_o \in V_o$ is denoted as v_o^i, $i = 1, 2$, with dashed edges $(v_o, v_o^i) \in E_o$. Each v_o^i represents the root of an individual surrogate decision sub-tree for model i. The height of a JST is the maximum number of decision edges (solid edges) in any root-to-leaf path.

Formally, a diff region is defined by the non-empty intersection of path-conditions of differently labelled leaves l_1, l_2 from two decision sub-trees rooted at the same or-node v_o. The collection of all diff regions specifies the diff ruleset:

$$\mathcal{R} = \left\{ \mathrm{pc}(l_1) \wedge \mathrm{pc}(l_2) : l_i \in \mathrm{leaves}(v_o^i),\ i = 1, 2, \right. \\ \left. \mathrm{label}(l_1) \neq \mathrm{label}(l_2),\ v_o \in V_o \right\}. \tag{3}$$

Construction The objective of JST construction is twofold: (a) *Maximize comparability*: To achieve maximal sharing of split conditions between the two decision tree surrogates, and (b) *Interpretability*: Achieve the above objective under the constraint of interpretability. We have chosen the height of the JST as the interpretability constraint.

The construction of a JST corresponding to the inputs M_1, M_2, X starts with evaluating $y_1 = M_1(X)$ and $y_2 = M_2(X)$. Starting from the root, at each internal node $v \in V_{dt}$, with inputs $(X_v, y_{1v} = M_1(X_v), y_{2v} = M_2(X_v))$ filtered by the node's path condition, the key choice is whether to create a joint decision node or an or-node for the surrogates to *diverge*. The choice of node type signifies whether the two surrogates will continue to share their split conditions or not. Once divergence happens at an or-node, the two sub-trees rooted at the or-node do not share any split nodes thereafter. Below we present a general condition for divergence and a simplified one implemented in our experiments.

In general, a divergence condition should compare the cost of a joint split to that of separate splits for the two models. In the context of greedy decision tree algorithms considered in this work, the comparison is between the sum of impurities for the best possible common split,

$$\operatorname{imp}(X_v, y_{1v}, y_{2v}) = \min_{\{f \in [d],\, t \in X_v[f]\}} H(f, t, X_v, y_{1v}) + H(f, t, X_v, y_{2v}), \quad (4)$$

and the impurities $\operatorname{imp}(X_v, y_{1v})$, $\operatorname{imp}(X_v, y_{2v})$ (2) for the best separate splits. One condition for divergence is

$$\operatorname{imp}(X_v, y_{1v}) + \operatorname{imp}(X_v, y_{2v}) \leq \alpha \operatorname{imp}(X_v, y_{1v}, y_{2v}) \quad (5)$$

for some $\alpha \leq 1$. The choice $\alpha = 1$ always results in divergence and thus reduces to the separate surrogate approach in Section 3. This happens because the left-hand side of (5) corresponds to separately minimizing the two terms in (4), hence ensuring that (5) is true. As α decreases, joint splits are favored. For $\alpha < 0$, divergence essentially never occurs.[1]

For this work, we choose to heavily bias the algorithm toward joint splits and greater interpretability of the resulting JST. In this case, we use the simplified condition

$$\operatorname{imp}(X_v, y_{1v}) = 0 \vee \operatorname{imp}(X_v, y_{2v}) = 0, \quad (6)$$

which results in divergence if at least one of the minimum impurity values is zero. The advantage of (6) over (5) is that the minimization in (4) to compute $\operatorname{imp}(X_v, y_{1v}, y_{2v})$ can be done lazily, only if (6) is not satisfied. If condition (6) is met, we create an or-node, two or-edges, and grow individual surrogate trees from that point onward. Figure 2 shows 1 instance (node v_{o0}) where (6) is met. A special case of (6) occurs when at least one of y_{1v}, y_{2v} contains only one label, i.e., it is already pure without splitting. The node v_{o1} in Figure 2 shows one such case.

The JST construction ends if pure leaf nodes are found or the height of the JST has reached a pre-defined hyper-parameter value k.

JST Refinement We now present an iterative process for refinement aimed at increasing precision of diff regions.

For each leaf l_i contributing to a diff region (3), if its samples (satisfying $\operatorname{pc}(l_i)$) have more than one label as given by the model M_i being approximated (the leaf is impure), we can further split it into two leaf nodes. This refines the decision tree surrogates only in the diff regions and not at all impure leaves. Next, diff regions are recomputed with the resulting leaf nodes. This process can continue for a pre-defined number of steps or until some budget is met. Every such iteration increases the tree depth by 1 (but not uniformly) and improves the fidelity of the individual subtree rooted at an or-node.

5 EXPERIMENTAL RESULTS

We report experimental results comparing the proposed IMD technique to learning separate surrogates for the two models (Section 5.1), and to direct difference modelling and the prior work of Nair et al. [2021] (Section 5.2). The effect of refinement is demonstrated in Section 5.3. The following paragraphs describe the setup of the experiments.

Datasets We have used 13 publicly available [Dua and Graff, 2017, Vanschoren et al., 2013, Alcalá-Fdez et al., 2011] tabular classification datasets, including both binary and multiclass classification tasks. As preprocessing steps, we dropped duplicate instances occurring in the original data, and one-hot encoded categorical features.

Models We split each dataset in the standard 70/30 ratio, and trained an array of machine learning models — Decision Tree Classifier (DT), Random Forest Classifier (RF), K-Neighbours Classifier (KN), Logistic Regression (LR), Gradient Boosting (GB), Multi-Layered Perceptron (MLP), and Gaussian Naive Bayes (GNB). For some models, multiple instances were trained with different parameter values. We have used the Scikit-learn [Pedregosa et al., 2011] implementations for training. Once trained, we did not do any performance tuning of the models, and used them as black boxes (through the `predict()` method only) for subsequent analyses. The dataset and model details including test set accuracies are reported in the supplementary material (SM).

Set Up We have selected two pairs of models per dataset corresponding to the largest and smallest differences in accuracy on the test set (indicated as *max* M_1-M_2 and *min* M_1-M_2 in Table 1). This ensures we compare models with contrasting predictive performance, as well as models that achieve similar accuracy. For fitting and evaluating diff models, including our IMD approach as well as baselines, we split the available dataset X (without labels) in a 70/30 ratio into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$. This split is not and does not have to be the same as the train/test splits for training and evaluating the underlying models. We perform 5 train/test splits and report in the main paper the mean of the following metrics across the 5 runs, with standard deviation values in the SM.

Metrics To measure how accurately we capture the true regions of disagreement between models M_1 and M_2, we use the following metrics. Given a test set $\mathcal{D}_{\text{test}}$, we have a subset of *true diff samples*:

$$\mathcal{T}_{\text{true}} = \{x \in \mathcal{D}_{\text{test}} \mid M_1(x) \neq M_2(x)\},$$

and the predicted diff samples by the diff model $\hat{D}(x)$:

$$\mathcal{T}_{\text{pred}} = \{x \in \mathcal{D}_{\text{test}} \mid \hat{D}(x) = 1\}.$$

Recall that in the case where we have extracted a diff ruleset $\mathcal{R}$ for $\hat{D}(x)$, $x \in \mathcal{T}_{\text{pred}}$ if there exists a rule $r \in \mathcal{R}$ that is satisfied by x.

[1] If $\operatorname{imp}(X_v, y_{1v}, y_{2v}) = 0$, then $\operatorname{imp}(X_v, y_{1v}) = \operatorname{imp}(X_v, y_{2v}) = 0$ also and the same (f, t) pair minimizes all three impurities. Hence divergence has no effect.

Precision (Pr) is the ratio $\frac{|\mathcal{T}_{\text{true}} \cap \mathcal{T}_{\text{pred}}|}{|\mathcal{T}_{\text{pred}}|}$, measuring the fraction of predicted diff samples that are true diff samples on the test set $\mathcal{D}_{\text{test}}$.

Recall (Re) is the ratio $\frac{|\mathcal{T}_{\text{true}} \cap \mathcal{T}_{\text{pred}}|}{|\mathcal{T}_{\text{true}}|}$, measuring the fraction of true diff samples in $\mathcal{D}_{\text{test}}$ that are correctly predicted.

Interpretability For interpretable diff models for which we have extracted a diff ruleset $\mathcal{R}$, we measure its interpretability in terms of the number of rules **(# r)** in the set, and the number of unique predicates **(# p)** summed over all the rules in the set. The choice of the above metrics is motivated by the works of Lakkaraju et al. [2016], Dash et al. [2018], Letham et al. [2015].

5.1 IMD AGAINST SEPARATE SURROGATES

First we study the effect of jointly training surrogates in IMD, which encourages sharing of split nodes, against training separate surrogates for the two models. Since these are both surrogate-based approaches to obtain a diff model $\hat{D}$, we compare the metrics for the *diff rulesets* extracted (as described in Section 3) from the surrogates. IMD extracts diff rulesets from JSTs, while the separate surrogate approach is a special case of IMD corresponding to $\alpha = 1$. The height (a.k.a. maximum depth) of the surrogates is restricted to 6 for both of the approaches. We do not perform the refinement step here as we study it in Section 5.3.

Observations The metrics are reported for 8 datasets in Table 1 (full version in Appendix). The differences in Pr, Re, and # rules are also tabulated for better readability. We also report the fraction of diff samples in $\mathcal{D}_{\text{test}}$ for each dataset and model pair combination in the "diffs" column. This value is also the precision of a trivial diff-model ($\hat{D}(x) = 1\ \forall x$, recall= 1.0), or any diff-model that predicts *diff* with probability q (recall= q), e.g., $q = 0.5$ is a random guesser. Clearly, diff prediction quality for both approaches is significantly better than random guessing.

To summarize the table, below we compare the approaches on the basis of average percentage increase or decrease in precision and recall (on going from separate to IMD) across all datasets. We also perform Wilcoxon's signed rank test (as recommended by Benavoli et al. [2016]) to verify the statistical significance of the observed differences.

For precision, we observe a very small drop (1.55% on average) going from separate surrogates to IMD. Wilcoxon's test's p-value is 0.269, implying no significant difference (at level 0.05) between the approaches. For recall, we observe that IMD has 23.45% poorer recall. Wilcoxon's test confirms this difference with a p-value of 0.0002, and a sign test also shows that separate surrogates have higher values of recall for 22 of the 26 benchmarks.

For the interpretability metrics however, IMD is the clear winner looking at the columns corresponding to numbers of rules (# r) and unique predicates (# p, in Appendix). If we simply average the numbers of rules and predicates to understand the scale of the difference (with the caveat that different datasets and model pairs have different complexities), the average number of rules for separate and IMD are 337.25 and 20.94, and the average numbers of predicates are 135.41 and 56.10. The corresponding p-values are also very low (on the order of 10^{-6}).

5.2 COMPARISON WITH OTHER APPROACHES

In this experiment, we compare the quality of prediction of the true dissimilarity D with respect to other baselines. The first two baselines are direct approaches (introduced in Section 3) as they relabel the instances as *diff*("1") or *non-diff*("0") and directly fit a classification model on the relabeled instances. Out of a huge number of possible models for this binary classification problem of predicting *diff* or *non-diff*, we choose Decision Tree (with `max_depth=6`) to be directly comparable to JST, and Gradient Boosting Classifier (`max_depth=6`, rest default settings in Scikit-learn) to provide a more expressive but uninterpretable benchmark. These choices are made to compare the quality of surrogate-based diff regions against directly modelled diff regions, and also to understand if we are significantly compromising on quality by not using a more expressive or uninterpretable model. As a third baseline, we compare to diff rulesets obtained from Grounded BRCG [Nair et al., 2021] ruleset surrogates for the two models. The surrogate-based approaches from the previous subsection, IMD (without refinement) and separate, are also included for completeness.

Observations We have listed the F1-scores (harmonic mean of precision and recall) in Table 2, and omitted the M_1 vs. M_2 column (same as in Table 1) for brevity. Since *BRCG Diff.* applies only to binary classification tasks, we only show it for those. Note that for IMD and separate surrogates, the precision and recall values are already reported in Table 1. For the other methods and datasets, precision, recall, and # rules (if applicable) are in the Appendix. On average, we observe that IMD achieves a 89.76% improvement in F1-score over Direct DT, and 98.52% improvement over the BRCG Diff. approach.[2] On the other hand, we do not observe a large drop in F1-score from the uninterpretable Direct GB to IMD (-5.87%). Similarly, the precision and recall differences in Section 5.1 combine to give a -15.26% decrease in going from separate surrogates to IMD.

We report mean ranks in Table 2 and performed Friedman's test following Demšar [2006], which confirms significant differences between the methods with a p-value on the order of 0.0006. Next we perform pairwise comparisons of IMD against the other approaches. The p-values from

[2] This is computed by removing the second subrow for tictactoe as F1 score is 0 for both Direct DT and BRCG Diff. and the jump is infinite. This removal is thus favorable toward them.

Table 1: Sep. surrogates shows slightly higher recall, but IMD shows comparable performance with much less complexity.

			Separate Surrogates			**IMD**			**Sep. − IMD**		
Dataset	M_1 **vs.** M_2	**diffs**	Pr	Re	#r	Pr	Re	#r	ΔPr	ΔRe	Δ#r
adult	max MLP1-GB	0.20	0.96	0.88	70.0	0.96	0.88	18.0	−0.00	−0.00	−52.0
	min MLP2-DT2	0.08	0.45	0.29	155.4	0.46	0.16	17.4	+0.01	−0.13	−138.0
bankm	max MLP2-GB	0.26	0.66	0.75	263.6	0.70	0.67	23.0	+0.04	−0.08	−240.6
	min MLP1-GNB	0.26	0.74	0.75	345.0	0.71	0.69	34.4	−0.03	−0.06	−310.6
eye	max RF1-GNB	0.56	0.65	0.66	1054.0	0.60	0.71	36.2	−0.06	+0.05	−1017.8
	min LR-MLP1	0.34	0.59	0.53	781.6	0.57	0.39	28.4	−0.02	−0.14	−753.2
heloc	max KN1-RF2	0.23	0.40	0.23	373.0	0.40	0.13	15.8	+0.00	−0.10	−357.2
	min GB-RF1	0.17	0.30	0.19	234.4	0.25	0.06	14.6	−0.05	−0.13	−219.8
magic	max RF1-GNB	0.25	0.75	0.58	362.8	0.75	0.52	25.0	+0.00	−0.06	−337.8
	min MLP2-DT2	0.11	0.43	0.36	282.6	0.42	0.17	11.0	−0.01	−0.18	−271.6
redwine	max RF1-KN2	0.37	0.46	0.52	627.8	0.52	0.25	29.0	+0.06	−0.27	−598.8
	min KN1-GNB	0.52	0.70	0.59	563.6	0.69	0.47	40.4	−0.01	−0.11	−523.2
tictactoe	max LR-GNB	0.34	0.76	0.78	109.6	0.76	0.89	24.4	−0.00	+0.11	−85.2
	min DT2-KN2	0.06	0.10	0.15	54.0	0.16	0.11	5.8	+0.05	−0.04	−48.2
waveform	max LR-DT1	0.18	0.45	0.52	746.0	0.49	0.27	33.2	+0.04	−0.25	−712.8
	min MLP1-RF2	0.11	0.17	0.32	725.0	0.10	0.02	9.0	−0.07	−0.30	−716.0

Wilcoxon's signed rank test are 0.00025, 0.043, 0.043, and 0.1594 against separate, *BRCG Diff.*, *Direct DT*, and *Direct GB* respectively. We pit these against the Holm-corrected thresholds of 0.0125, 0.017, 0.025, 0.05, and observe that only the first one (IMD vs. separate) is significant for this set of values. However, we emphasize that although separate and Direct GB have consistently higher F1-scores than IMD, the size of the differences is small and IMD is considerably more interpretable. For the interpretable methods, the average numbers of rules observed for IMD, Direct DT, and BRCG Diff. are 16.05, 10.50, 37.69 (separate surrogates was already discussed in Section 5.1).

We present further experiments (in Appendix) varying the depth to understand the accuracy-complexity trade-off for Direct DT, Separate and IMD extensively. While the tradeoffs for Direct DT and IMD are competitive, both of them are consistently better than Separate. We also discuss qualitative comparison between Direct DT and IMD which brings out the benefit of IMD in placing the diff rules in the context of the models' decision logic, as already seen in Figure 2.

5.3 EFFECT OF REFINEMENT

To investigate the effect of the refinement step of IMD (described at the end of Section 4.1), we compare diff rulesets obtained from three variations of the algorithm — IMD with maximum depth of 6 (IMD$_6$), same as in previous experiments; IMD$_6$ with 1 iteration of refinement (IMD$_{6+1}$); and IMD with maximum depth of 7 (IMD$_7$).

Looking at Table 3 (all benchmarks not shown for lack of space), we observe improvement in precision from IMD$_6$ to IMD$_{6+1}$ (11.27% on average), and interestingly, also from IMD$_7$ to IMD$_{6+1}$ (4.22% on average). The p-values from Wilcoxon's test are on the order of 10^{-3} for both comparisons, validating the significance of the improvement. The average numbers of rules for the three approaches are 20.93, 28.77, and 41.01 respectively, confirming that IMD$_{6+1}$ only refines selectively compared to IMD$_7$.

The results demonstrate that selective splitting of impure leaf nodes only in predicted diff regions (IMD$_{6+1}$), improves precision compared to regular tree splitting of *all* impure nodes (IMD$_7$). However, this improvement is to be taken with some caution as it comes at the cost of a consistent drop in recall (15.37% from IMD$_6$ and 25.14% from IMD$_7$ averaged across all benchmarks). Thus we recommend refinement specifically for scenarios requiring high precision difference modelling.

Experimental Conclusions IMD has close to the same F1-scores as the top methods in our comparison, separate surrogates and the (uninterpretable) Direct GB. At the same time, IMD yields much more concise results, with an order of magnitude fewer diff rules than separate surrogates. This affirms the benefit of sharing nodes in JST, which localizes differences before divergence. We also see (in SM) how the features deemed important by JST are close to what the models also use in their decision logic via feature importance computations. This establishes our claim that JSTs are able to achieve two things at once: interpretable surrogates that can be compared easily for the two models. Refinement further improves the precision of IMD, but at the cost of recall and interpretability. Additional experiments (in SM) also support these conclusions.

Table 2: Comparison of F1-scores. The mean ranks (↓ the better) highlight that sep. surr., and Direct GB are most accurate, but IMD is close with greater interpretability.

		Sep.	**Direct**	**Direct**	**BRCG**
Dataset	**IMD**	**Surr.**	**DT**	**GB**	**Diff.**
adult	0.92	0.92	0.92	0.98	0.33
	0.23	0.34	0.17	0.61	0.31
bankm	0.68	0.70	0.69	0.77	0.41
	0.70	0.75	0.68	0.82	0.41
banknote	0.89	0.89	0.83	0.94	0.27
	0.52	0.56	0.57	0.63	0.06
bc	0.39	0.41	0.17	0.00	0.10
	0.25	0.37	0.28	0.19	0.13
diabetes	0.32	0.43	0.21	0.35	0.35
	0.32	0.41	0.09	0.22	0.30
heloc	0.19	0.29	0.03	0.14	0.37
	0.10	0.22	0.02	0.05	0.27
magic	0.62	0.65	0.63	0.78	0.40
	0.24	0.39	0.14	0.27	0.20
mushroom	0.75	0.80	0.81	0.97	0.76
	0.72	0.80	0.81	0.97	0.74
tictactoe	0.82	0.77	0.77	0.82	0.83
	0.12	0.12	0.00	0.09	0.00
mean rank	**3.278**	**2.056**	**3.694**	**2.278**	**3.694**

5.4 CASE STUDY

We conclude by demonstrating a practical application of the method in the fairness area in the advertising domain. Bias in ad campaigns leads to poor outcomes for companies not reaching the right audience, and for customers who are incorrectly targeted. Bias mitigation aims to correct this by changing models to have more equitable outcomes.

Our IMD method can be used to assess the impact of bias mitigation on a model. In this case study, a bias mitigation method was applied to the group of *non*-homeowners who had higher predicted rates of conversion (relative to ground truth). The root node of the JST captures this group. Figure 3 shows a part of the JST (full tree in the Appendix). Although the non-homeowner group is already over-predicted, the JST shows that for certain cohorts within the group (those outside the ages of 25-34), conversions are predicted where the model before mitigation would not have. Interpretable model differencing here captures unintended consequences of model alterations.

6 CONCLUSION

We addressed the problem of interpretable model differencing, localizing and representing differences between ML models for the same task. We proposed JST to provide a unified view of the similarities and dissimilarities between the models as well as a succinct ruleset representation. Experimental results indicate that the proposed IMD approach yields a favorable trade-off between accuracy and interpretability in predicting model differences.

Table 3: Precision improves on refinement (IMD_{6+1}).

Dataset	**IMD_6**	**IMD_{6+1}**	**IMD_7**
adult	0.96	**0.96**	0.95
	0.46	**0.59**	0.53
bankm	0.70	**0.78**	0.77
	0.71	**0.79**	0.74
eye	0.60	**0.67**	0.62
	0.57	**0.64**	0.57
heloc	0.40	**0.45**	0.42
	0.25	0.25	**0.26**
magic	0.75	**0.80**	0.73
	0.42	**0.55**	0.46
redwine	0.52	**0.56**	0.48
	0.69	**0.73**	0.68
tictactoe	0.76	**0.79**	0.78
	0.16	**0.19**	0.18
waveform	0.49	**0.54**	0.49
	0.10	0.14	**0.17**

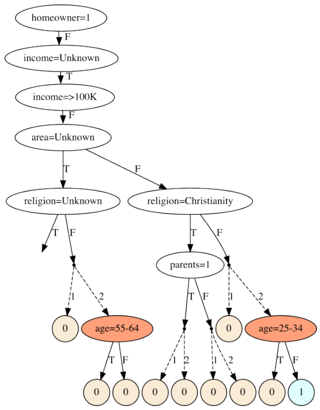

Figure 3: A subtree of the JST showing an unintended increase in predicted conversions after bias mitigation for a cohort of the already over-predicted group of non-homeowners.

The current work is limited to comparing classifiers in terms of 0-1 dissimilarity. Since IMD is based on decision trees, its interpretability is limited to domains where the features are interpretable. While we have chosen to extend greedy decision tree algorithms due to ease and scalability, the resulting JSTs accordingly have no guarantees of optimality.

Future work could seek to address the above limitations. To extend the framework to regression tasks, a potential avenue is to threshold the difference function $D(M_1(x), M_2(x))$ and apply the classification framework presented herein. The problem of interpretable model differencing for images and language remains open. The constituent features for these modalities are generally not interpretable making the diff rulesets uninterpretable without additional considerations.

7 ACKNOWLEDGEMENTS

This work was partially funded by the European Union's Horizon Europe research and innovation programme under grant agreement no. 101070568.

References

Sina Aghaei, Mohammad Javad Azizi, and Phebe Vayanos. Learning optimal and fair decision trees for non-discriminative decision-making. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1418–1426, 2019.

Jesús Alcalá-Fdez, Alberto Fernández, Julián Luengo, Joaquín Derrac, Salvador García, Luciano Sánchez, and Francisco Herrera. Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic & Soft Computing*, 17, 2011.

Maksym Andriushchenko and Matthias Hein. Provably robust boosted decision stumps and trees against adversarial attacks. *Advances in Neural Information Processing Systems*, 32, 2019.

Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2654–2662, 2014. URL `http://papers.nips.cc/paper/5484-do-deep-nets-really-need-to-be-deep.pdf`.

Gagan Bansal, Besmira Nushi, Ece Kamar, Walter S Lasecki, Daniel S Weld, and Eric Horvitz. Beyond accuracy: The role of mental models in human-ai team performance. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 7, pages 2–11, 2019.

Osbert Bastani, Carolyn Kim, and Hamsa Bastani. Interpretability via model extraction. In *Workshop on Fairness, Accountability, and Transparency in Machine Learning (FAT/ML)*, 2017. URL `https://arxiv.org/abs/1706.09773`.

Alessio Benavoli, Giorgio Corani, and Francesca Mangili. Should we really use post-hoc tests based on mean-ranks? *Journal of Machine Learning Research*, 17(5):1–10, 2016. URL `http://jmlr.org/papers/v17/benavoli16a.html`.

Y. Bu, J. Lu, and V. V. Veeravalli. Model change detection with application to machine learning. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5341–5346, 2019.

Cristian Bucilă, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, page 535–541, 2006. URL `https://doi.org/10.1145/1150402.1150464`.

Hongge Chen, Huan Zhang, Duane Boning, and Cho-Jui Hsieh. Robust decision trees against adversarial examples. In *International Conference on Machine Learning*, pages 1122–1131. PMLR, 2019.

Alexandra Chouldechova and Max G'Sell. Fairer and more accurate, but for whom? *arXiv preprint arXiv:1707.00046*, 2017.

Jürgen Cito, Isil Dillig, Seohyun Kim, Vijayaraghavan Murali, and Satish Chandra. Explaining mispredictions of machine learning models using rule induction. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 716–727, 2021.

Mark William Craven. *Extracting Comprehensible Models from Trained Neural Networks*. PhD thesis, 1996. AAI9700774.

Sanjeeb Dash, Oktay Gunluk, and Dennis Wei. Boolean decision rules via column generation. *Advances in neural information processing systems*, 31, 2018.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(1):1–30, 2006. URL `http://jmlr.org/papers/v7/demsar06a.html`.

Jaka Demšar and Zoran Bosnić. Detecting concept drift in data streams using model explanation. *Expert Syst. Appl.*, 92:546–559, 2018.

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

Christopher Duckworth, Francis P Chmiel, Dan K Burns, Zlatko D Zlatev, Neil M White, Thomas WV Daniels, Michael Kiuber, and Michael J Boniface. Using explainable machine learning to characterise data drift and detect emergent health risks for emergency department admissions during covid-19. *Scientific reports*, 11(1):1–10, 2021.

Alex A Freitas. Comprehensible classification models: a position paper. *ACM SIGKDD explorations newsletter*, 15(1):1–10, 2014.

Nicholas Frosst and Geoffrey Hinton. Distilling a neural network into a soft decision tree. In *Comprehensibility and Explanation in AI and ML (CEX) Workshop, 16th International Conference of the Italian Association for Artificial Intelligence (AI*IA)*, 2017. URL `https://arxiv.org/pdf/1711.09784.pdf`.

Jun Geng, Bingwen Zhang, Lauren M. Huie, and Lifeng Lai. Online change-point detection of linear regression models. *IEEE Transactions on Signal Processing*, 67(12): 3316–3329, 2019.

Maayan Harel, Koby Crammer, Ran El-Yaniv, and Shie Mannor. Concept drift detection through resampling. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, page II–1009–II–1017. JMLR.org, 2014.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. URL `http://arxiv.org/abs/1503.02531`.

Todd Kulesza, Simone Stumpf, Margaret Burnett, and Irwin Kwan. Tell me more? the effects of mental model soundness on personalizing an intelligent agent. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1–10, 2012.

Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *International Conference on Knowledge Discovery and Data Mining*, pages 1675–1684, 2016.

Benjamin Letham, Cynthia Rudin, Tyler H McCormick, and David Madigan. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3):1350–1371, 2015.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations (ICLR)*, 2016. URL `http://leon.bottou.org/papers/lopez-paz-2016`.

Michal Moshkovitz, Yao-Yuan Yang, and Kamalika Chaudhuri. Connecting interpretability and robustness in decision trees through separation. In *International Conference on Machine Learning*, pages 7839–7849. PMLR, 2021.

Rahul Nair, Massimiliano Mattetti, Elizabeth Daly, Dennis Wei, Oznur Alkan, and Yunfeng Zhang. What changed? interpretable model comparison. IJCAI, 2021.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

JR Quinlan. Induction of decision trees. mach. learn. 1986.

Sarah Tan, Rich Caruana, Giles Hooker, and Yin Lou. Distill-and-compare: Auditing black-box models using transparent model distillation. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, page 303–310, 2018. URL `https://doi.org/10.1145/3278721.3278725`.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. Openml: networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013. doi: 10.1145/2641190.2641198. URL `http://doi.acm.org/10.1145/2641190.264119`.