Robustness Quantification for Discriminative Models: A New Robustness Metric and its Application to Dynamic Classifier Selection

Rodrigo F. L. Lassance[1,2,3] **Jasper De Bock**[1]

[1]Foundations Lab for imprecise probabilities (FLip), Ghent University, Ghent, Belgium
[2]Statistics Dept., Federal University of São Carlos, São Carlos, São Paulo, Brazil
[3]Institute of Mathematic and Computer Sciences, University of São Paulo, São Carlos, São Paulo, Brazil

Abstract

Among the different possible strategies for evaluating the reliability of individual predictions of classifiers, robustness quantification stands out as a method that evaluates how much uncertainty a classifier could cope with before changing its prediction. However, its applicability is more limited than some of its alternatives, since it requires the use of generative models and restricts the analyses either to specific model architectures or discrete features. In this work, we propose a new robustness metric applicable to any probabilistic discriminative classifier and any type of features. We demonstrate that this new metric is capable of distinguishing between reliable and unreliable predictions, and use this observation to develop new strategies for dynamic classifier selection.

1 INTRODUCTION

Machine learning models possess great predictive capacity, but this capacity comes with a level of unreliability that is hard to assess. From a modeling perspective alone, the majority of the methods used to make high-stakes decisions lack transparency in what their true decision process is, making their users potentially subject to harmful consequences in the process [O'Neil, 2016]. Although there have been attempts to make such black box models more interpretable [Molnar, 2025] or to switch to models that are inherently interpretable instead [Rudin, 2019], these contributions remain insufficient due to reasons that go beyond the models themselves. After all, a model is only as good as the data that has been provided to it, with the inherent variability of the phenomena and the lack of all meaningful features leading to an unreliability source referred to as Aleatoric Uncertainty (AU). Moreover, when **(i)** the data is not sufficient for the model to differentiate pattern from noise to an acceptable degree or **(ii)** there is a discrepancy between the data being used to train the model and the context in which the model will be applied to, this also leads to another unreliability source: Epistemic Uncertainty (EU).

While completely removing these sources of uncertainty is impossible by definition, progress has been made in evaluating how reliable an individual prediction is in the field of uncertainty quantification. Many different approaches have already been explored [Hüllermeier and Waegeman, 2021], with the attempt of estimating and separating AU and EU among the most popular strategies. If successfully done, this allows decision makers to refrain from trusting the outputs of a model in specific circumstances, avoiding the more undesirable repercussions that could come. However, properly quantifying EU is rarely feasible from the data used for training alone, since it most often diverges from the true application that a practitioner has in mind. Aside from specific situations, some aspects of EU are fundamentally inaccessible in practice.

A promising alternative to directly estimating EU is using robustness quantification. Similarly to uncertainty quantification, earlier work [Detavernier and De Bock, 2025a,b] has demonstrated that robustness quantification can be used to assess the reliability of individual predictions. Unlike uncertainty metrics, which aim to quantify the amount of uncertainty that influences the decision, robustness metrics aim to quantify how uncertain one can be about the model while still being able to trust its decision, thereby sidestepping the problem of estimating the amount of uncertainty there actually is. Both types of metrics tend to correlate with the accuracy of the predictions, and their relative performance varies with the context. Robustness metrics tend to perform best exactly in the contexts where uncertainty is difficult to estimate, such as in the presence of distribution shift, model misspecification, or small data regimes.

Compared to other applications of the term "robustness" in machine learning, one of the differentiating aspects of robustness quantification is that it is an instance-based metric,

i.e., it is an assessment of how reliable a specific individual prediction is based on the underlying model and the values of the features used as input. It is also different from the notion of adversarial robustness [Muhammad and Bae, 2022], which focuses on perturbing the features and is mainly used in applications with image data. Robustness quantification, on the other hand, comes from a notion of perturbing the joint distributions of the trained model, which is more in line with ideas in imprecise probabilities [Augustin et al., 2014].

Important downsides of robustness quantification, at this point, are that it can only be applied to generative models, is mainly based on epsilon-contamination, and is restricted to either specific model architectures [De Bock et al., 2014, Correia and de Campos, 2019, Correia et al., 2020] or to fully discrete sets of features [Detavernier and De Bock, 2025a]. In this work, we improve on these current limitations by proposing a new robustness metric that is applicable to any probabilistic discriminative classifier (section 2). Instead of using epsilon-contamination, our metric is based on the Constant Odds Ratio (COR) perturbation, yielding a metric that is not restricted to discrete features. We start in Section 2 with an introduction to robustness quantification and explain how, in particular, a dissimilarity function can be used to define a robustness metric. Theorem 1 provides a closed-form expression for the COR perturbation case that we focus on and is followed by a list of practical considerations that motivate this particular choice of perturbation. Section 3 goes on to test the performance of this new metric. In a first batch of experiments, we use Accuracy Rejection Curves to demonstrate that this metric correlates nicely with accuracy, that it does this better than an alternative competitor, and that it can do this for several model architectures (section 3.1). Our main application uses robustness to perform dynamic selection of classifiers, offering two strategies for choosing which model to use as a predictor given a set of features (section 3.2). We end in Section 4 with a discussion that highlights possible venues for future research.

2 ROBUSTNESS QUANTIFICATION

Let $Y \in \mathcal{Y}$ be a discrete class variable and $X \in \mathcal{X}$ its vector of features. The features can be purely discrete, purely continuous or a mixture of both. Uncertainty about (Y, X) is expressed by a probability measure P on a suitable sigma algebra $\mathcal{A}$ of subsets of $\mathcal{Y} \times \mathcal{X}$ (e.g. the power set when X is purely discrete or a product of a power set and Borel sigma algebra if X purely continuous or mixed). We furthermore assume that P is absolutely continuous w.r.t. an adequate base measure μ (e.g., the counting measure when X is purely discrete or the product of a counting and Lebesgue measure when X is purely continuous or mixed), which guarantees that P has a density $p = {}^{dP}/_{d\mu}$ w.r.t. this base measure μ. We denote the set of all such probability measures by $\mathbb{P}(\mathcal{Y}, \mathcal{X})$.

We consider a classification problem where the goal is to predict the value of Y given x based on P or, equivalently, based on p. The classifier that minimizes the 0-1 loss then predicts the most likely class given the features x, which is given by

$$\begin{aligned} g_p(x) :&= \arg\max_{y \in \mathcal{Y}} p(y|x) = \arg\max_{y \in \mathcal{Y}} \frac{p(y,x)}{p(x)} \\ &= \arg\max_{y \in \mathcal{Y}} p(y,x). \end{aligned} \tag{1}$$

We are interested in quantifying the robustness of this prediction, by determining how stable it is with respect to divergences from the original measure P. To this end, following Detavernier and De Bock [2025a], we consider parametrized perturbations $\mathcal{P}_\delta$ around P, with $\delta \in \mathbb{R}_{\geq 0}$. We call the prediction $g_p(x)$ robust w.r.t. $\mathcal{P}_\delta$ if, for all $P' \in \mathcal{P}_\delta$, the prediction $g_{p'}(x)$ is the same as $g_p(x)$. The robustness $r(x)$ of $g_p(x)$ is then quantified as the largest δ such that $g_p(x)$ is robust w.r.t. $\mathcal{P}_\delta$.

Work on robustness quantification has so far mainly focused on perturbations $\mathcal{P}_\delta$ based on epsilon-contamination. One approach consists in applying epsilon-contamination directly to the global model P, resulting in perturbations of the form $\mathcal{P}_\varepsilon = \{(1-\varepsilon)P + \varepsilon Q : Q \in \mathbb{P}(\mathcal{Y}, \mathcal{X})\}$ with $\varepsilon \in [0, 1]$. This leads to simple closed form expressions for the robustness metric $r(x)$, but it is only meaningful if X is discrete since it otherwise typically leads to robustness values of zero. Another approach consists in applying epsilon-contamination to the local parameters of specific parametric models such as Naive Bayes classifiers [Detavernier and De Bock, 2025a], Sum-Product Networks [Correia and de Campos, 2019] or Generative Forests [Correia et al., 2020]. This local approach remains meaningful for continuous or mixed features as well (at least for Sum-Product Networks and Generative Forests), but is only applicable to specific parametric models and can be computationally expensive.

In this work, we study robustness quantification based on a dissimilarity function d between probability measures. That is, we consider perturbations of the type

$$\mathcal{P}_\delta = \{Q \in \mathbb{P}(\mathcal{Y}, \mathcal{X}) : d(P, Q) \leq \delta\} \tag{2}$$

with $\delta \in \mathbb{R}_{\geq 0}$. In other words, the prediction is robust w.r.t. $\mathcal{P}_\delta$ if the prediction $g_q(x)$ is the same as $g_p(x)$ for all $Q \in \mathbb{P}(\mathcal{Y}, \mathcal{X})$ such that $d(P, Q) \leq \delta$, and the robustness $r(x)$ of the prediction $g_p(x)$ is the largest δ for which this is the case. To indicate the influence of the choice of d, we will write $r_d(x)$ instead of $r(x)$ when referring to the robustness metric based on a specific dissimilarity function d.

An interesting property of a robustness metric based on a dissimilarity function d is presented in the proposition below and follows by definition.

Proposition 1. *For any $Q \in \mathbb{P}(\mathcal{Y}, \mathcal{X})$ and $x \in \mathcal{X}$:*

$$d(P, Q) < r_d(x) \implies g_p(x) = g_q(x).$$

Notwithstanding the possibility of the measure P differing substantially from the true Data Generating Process (DGP) of the new data, Proposition 1 guarantees that their predictions will be the same for inputs whose robustness is high enough. Therefore, if the DGP provides reliable predictions, then so does P when robustness is high.

We focus in particular on the distance function d_{COR}, defined for all $P, Q \in \mathbb{P}(\mathcal{Y}, \mathcal{X})$ by

$$d_{\mathrm{COR}}(P, Q) := \sup_{\substack{A,B \in \mathcal{A} \\ Q(A)>0, P(B)>0}} \left(1 - \frac{P(A)Q(B)}{Q(A)P(B)}\right). \quad (3)$$

The idea is that P and Q are similar if the odds $P(A)/P(B)$ and $Q(A)/Q(B)$ are similar for any two events A and B. Similarly to epsilon-contamination, the perturbation induced by Equation (3) is related to imprecise probabilities. More specifically, Montes et al. [2020] shows that the perturbations that correspond to this distance function are Constant Odds Ratio (COR) models [Augustin et al., 2014, Section 4.7.2], which can also be given a behavioral interpretation in terms of gambling [Walley, 1991, Section 2.9.4].

A closely related dissimilarity function is d^*_{COR}, defined for all $P, Q \in \mathbb{P}(\mathcal{Y}, \mathcal{X})$ by

$$\begin{aligned} d^*_{\mathrm{COR}}(P, Q) :&= \sup_{\substack{A,B \in \mathcal{A} \\ Q(A)>0, P(B)>0}} \frac{P(A)Q(B)}{Q(A)P(B)} \\ &= 1 - \frac{1}{d_{\mathrm{COR}}(P, Q)}. \end{aligned} \quad (4)$$

In terms of the densities p and q, this simplifies to

$$\begin{aligned} d^*_{\mathrm{COR}}(P, Q) :&= \operatorname{ess\,sup} \frac{p}{q} \operatorname{ess\,sup} \frac{q}{p} \\ &= \frac{\operatorname{ess\,sup} \frac{q}{p}}{\operatorname{ess\,inf} \frac{q}{p}} = \frac{\operatorname{ess\,sup} L}{\operatorname{ess\,inf} L}, \end{aligned} \quad (5)$$

with

$$\begin{aligned} \operatorname{ess\,sup} f &:= \inf\{a \in \mathbb{R} : f \leq a \ \ \mu\text{-a.s.}\}, \\ \operatorname{ess\,inf} f &:= \sup\{a \in \mathbb{R} : f \geq a \ \ \mu\text{-a.s.}\} \end{aligned}$$

$L = \frac{dQ}{dP} = \frac{q}{p}$ the likelihood ratio of Q w.r.t. P [Dümbgen et al., 2021]. The $\operatorname{ess\,sup}$ (respectively $\operatorname{ess\,inf}$) is a version of the supremum (infimum) for functions that are only uniquely defined up to measure zero.

Since the distance function d_{COR} and dissimilarity function d^*_{COR} are monotone transformations of each other, the same is true for the resulting robustness metrics: for all $x \in \mathcal{X}$,

$$r_{d_{\mathrm{COR}}}(x) = 1 - \frac{1}{r_{d^*_{\mathrm{COR}}}(x)} \quad (6)$$

and

$$r_{d^*_{\mathrm{COR}}}(x) = \frac{1}{1 - r_{d_{\mathrm{COR}}}(x)}. \quad (7)$$

For this reason, we can equivalently work with either of these dissimilarity functions and their corresponding robustness metrics. We will work with d^*_{COR} and $r_{d^*_{\mathrm{COR}}}$ in our proofs, out of convenience, but will mostly focus on d_{COR} and $r_{d_{\mathrm{COR}}}$ in the remainder of the paper because we find $r_{d_{\mathrm{COR}}}$ more intuitive to interpret.

The following result, the proof of which is available in section 5, provides closed-form expressions for the robustness metrics $r_{d^*_{\mathrm{COR}}}$ and $r_{d_{\mathrm{COR}}}$.

Theorem 1. *Consider a measure P in $\mathcal{P}(\mathcal{Y}, \mathcal{X})$, its corresponding density p and, for any given features $x \in \mathcal{X}$, the most and second most likely class according to p given x:*

$$\hat{y}_1 \in \arg\max_{y \in \mathcal{Y}} p(y|x) \ \text{ and } \ \hat{y}_2 \in \arg\max_{y \in \mathcal{Y} \setminus \{\hat{y}_1\}} p(y|x).$$

Then the robustness of the prediction $\hat{y}_1$ w.r.t. d^*_{COR} and d_{COR} is given by, respectively,

$$r_{d^*_{\mathrm{COR}}}(x) = \frac{p(\hat{y}_1, x)}{p(\hat{y}_2, x)} \quad (8)$$

and

$$r_{d_{\mathrm{COR}}}(x) = \frac{p(\hat{y}_1, x) - p(\hat{y}_2, x)}{p(\hat{y}_1, x)}. \quad (9)$$

This result has a number of noteworthy consequences that are presented below. To the best of our knowledge, no other available robustness metric possesses all of these desirable properties simultaneously, further justifying their adoption.

Easy to compute. Since both metrics have simple closed-form expressions, this makes them particularly easy to evaluate, whereas existing local robustness metrics typically require performing optimization procedures [Correia and de Campos, 2019, Correia et al., 2020].

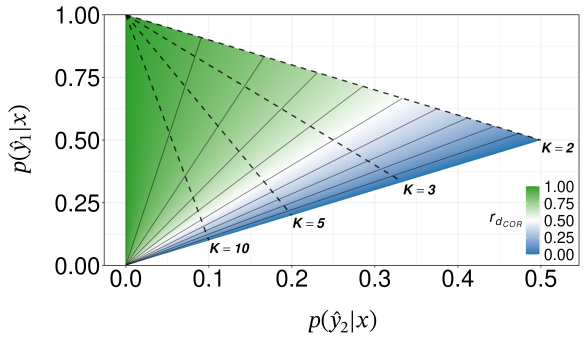
Applicable to any type of features. Unlike global robustness metrics based on epsilon-contamination, our metrics are meaningful (not automatically zero) regardless of whether the features are discrete, continuous or mixed.

Compatible with discriminative models. A simple application of Bayes' rule allows us to rewrite the obtained expressions in terms of conditional probabilities:

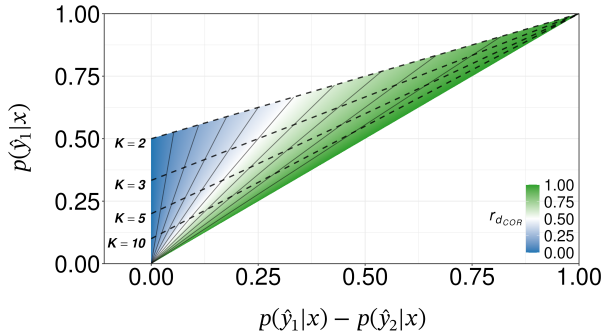
$$r_{d^*_{\mathrm{COR}}}(x) = \frac{p(\hat{y}_1|x)}{p(\hat{y}_2|x)} \quad (10)$$

and

$$r_{d_{\mathrm{COR}}}(x) = \frac{p(\hat{y}_1|x) - p(\hat{y}_2|x)}{p(\hat{y}_1|x)}. \quad (11)$$



(a) Probability of the second best class



(b) Difference of the top 2 probabilities

Figure 1: Contour plots of $r_{d_{\text{COR}}}$ as a function of to $p(\hat{y}_1|x)$ and some other variable. The dashed lines are the lower bounds for when Y has K categories.

This implies that these metrics can be applied to any machine learning architecture that leads to a discriminative model, that is, one for which only the conditional distribution $p(\cdot|x)$ is available. Our robustness metrics are the first in the literature endowed with such capability.

Easy to interpret. Since both expressions are based solely on the (conditional or joint) probability of the first and second most probable class, they are extremely simple to interpret, even without detailed knowledge about the reasoning that led to them. In particular, we personally very much like the interpretation of $r_{d_{\text{COR}}}$ as the probability difference between the first and second most probable class, relative to the absolute probability of the most probable one.

To provide some insight into the interpretation of our metric, Figure 1 presents two contour plots of $r_{d_{\text{COR}}}(x)$ based on the values of the top probability and either $p(\hat{y}_2|x)$ or $p(\hat{y}_1|x) - p(\hat{y}_2|x)$. The values one can choose for $p(\hat{y}_1|x)$ and the other entry are restricted by the number of classes in $\mathcal{Y}$, which we denote by K. For a fixed value of K, the figures present a dashed line associated to it, and any point in the colored region above this line obeys the restrictions imposed by the setting.

In Figure 1a, we make a direct study of how the top two probabilities affect robustness. As expected, robustness is high when $p(\hat{y}_1|x)$ and and $p(\hat{y}_2|x)$ are far apart, but how distant from each other they have to be depends on how low $p(\hat{y}_2|x)$ is: it is not the absolute difference that matters, but by how big a factor they differ. So even if $p(\hat{y}_1|x)$ is small, robustness can still be high provided $p(\hat{y}_2|x)$ is only a small fraction of $p(\hat{y}_1|x)$. This is only possible if K is sufficiently high though because $p(\hat{y}_2|x)$ being low implies that all other classes also have low probabilities associated to them, while the sum of all probabilities still needs to be one.

Another point of view is presented in Figure 1b, where we switch $p(\hat{y}_2|x)$ for $p(\hat{y}_1|x) - p(\hat{y}_2|x)$, making the figure more directly related to Equation (11) and to two metrics commonly used to evaluate the reliability of predictions: the maximum probability (y-axis) and the confidence gap (x-axis). While smaller values of $p(\hat{y}_1|x) - p(\hat{y}_2|x)$ lead to low robustness when K is small, this ceases to be the case for higher K as long as $p(\hat{y}_1|x)$ is close to the confidence gap (for points close to the diagonal). This demonstrates that our robustness metric can behave in fundamentally different ways depending on the number of classes in $\mathcal{Y}$, and that evaluating only $p(\hat{y}_1|x)$ or $p(\hat{y}_1|x) - p(\hat{y}_2|x)$ is insufficient for capturing this feature.

We end this section by adding a caveat to our robustness metric. Similarly to many of the metrics used in uncertainty quantification (see section 4 of Detavernier and De Bock [2025a] for some examples), $r_{d_{\text{COR}}}$ may favor overconfident predictions, where the model assigns high probability to a single class even though that is not reflected in the data. We therefore recommend to first apply procedures that enhance the overall reliability of the model whenever possible—such as calibration and evaluating overall performance—and to determine the robustness of the individual predictions of such a model in a secondary step.

3 EXPERIMENTS

The experiments undertaken in this work have two objectives: **(i)** demonstrate that our reliability metric(s) are capable of assessing the reliability of the predictions of classifiers (section 3.1) and **(ii)** offer one possible application in which we use the notion of robustness to develop new Dynamic Classifier Selection methods (section 3.2). Detailed information about all datasets in our experiments is given in Table 1; the sources were the OpenML [Vanschoren et al., 2014], UCI [Kelly et al., 2023] and PMLB [Romano et al., 2021] benchmark repositories. Code reproducing the results can be found in `https://github.com/rflassance/rob4discriminative`.

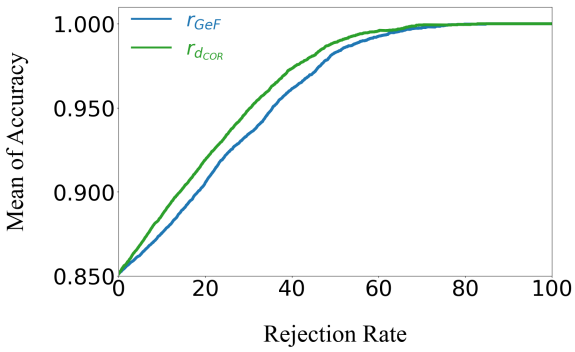
3.1 CORRELATION WITH ACCURACY

To assess the quality of our robustness metric(s), we follow Detavernier and De Bock [2025a,b] in using Accuracy Re-

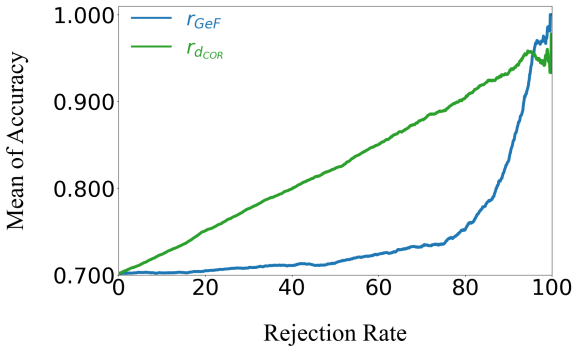
Table 1: Details about the datasets in our experiments.

Label	Name	n	#features (%cont.)	Source
D_1	authent	1372	4 (100%)	OpenML
D_2	bank-additional-full	41188	20 (25%)	OpenML
D_3	diabetes	768	8 (25%)	OpenML
D_4	electricity	45312	8 (88%)	OpenML
D_5	gesture	9873	32 (100%)	OpenML
D_6	magic	19020	10 (100%)	OpenML
D_7	robot	5456	24 (100%)	OpenML
D_8	segment	2310	19 (84%)	OpenML
D_9	students	4424	36 (19%)	UCI
D_{10}	texture	5500	40 (100%)	OpenML
D_{11}	twonorm	7400	20 (100%)	OpenML
D_{12}	vowel	990	12 (83%)	OpenML
D_{13}	waveform_21	5000	21 (100%)	PMLB
D_{14}	waveform_40	5000	40 (100%)	PMLB
D_{15}	wdbc	569	30 (100%)	OpenML

jection Curves [ARC, Nadeem et al., 2009]. To generate such an ARC, we first order the data according to a specific strategy, which in this case is in order of increasing values of robustness. Next, we evaluate the accuracy of the model in regards to the ordered data, gradually throwing away the first samples (with low robustness, in this case) and recalculating the accuracy for the remaining data, leading to an accuracy curve indexed by the proportion of samples that were removed (the rejection rate). Examples of such ARCs, which we'll analyse in the next section, can be seen in Figure 2. When the ordering criteria is good, the accuracy of the model quickly goes to 1 as we disregard samples. Conversely, when the ordering is not good, the ARC remains closer to the starting accuracy. This way, ARCs provide a visual method for evaluating how well a robustness metric is at correlating with accuracy. In sections 3.1.1 and 3.1.2, we perform 15 train-test splits for each dataset analyzed. Then, for every split, we obtain the ARC for the test set and present the average of the ARCs as our result.



(a) waveform_21 dataset (D_{13})



(b) students dataset (D_9)

Figure 2: Comparison between the ARCs of r_{GeF} and $r_{d_{COR}}$ for two datasets.

3.1.1 Comparison Between Robustness Metrics

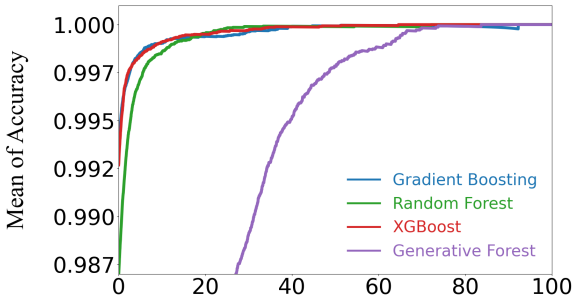
Our first experiment compares how our robustness metric, $r_{d_{COR}}$, fares when compared to an alternative. Considering that part of the focus of this work is to provide a robustness metric that remains meaningful when the datasets possess continuous features, we turn our attention to another method with a similar capability: a local robustness metric based on Generative Forests [GeF, Correia et al., 2020] that we refer to as r_{GeF}. This metric is specifically designed for GeFs— which are a generative version of Random Forests [Breiman, 2001]—and epsilon-contaminates the parameters of a Generative Forest to assess robustness.

Figure 2 shows the ARC curves of both robustness metrics for two different datasets, as an illustration of two extreme situations that occur throughout the datasets. In Figure 2a, we note that both robustness metrics provide a good ordering criteria for the data, with $r_{d_{COR}}$ slightly outperforming r_{GeF}. In the other extreme, Figure 2b highlights a situation in which r_{GeF} is doing a poor job at correlating with accuracy, whereas $r_{d_{COR}}$ remains more consistent. Considering that in both cases $r_{d_{COR}}$ has a better performance, and that we observed similar behaviour in the other datasets, this leads us to disregard the use of r_{GeF} in further analyses and focus solely on $r_{d_{COR}}$.

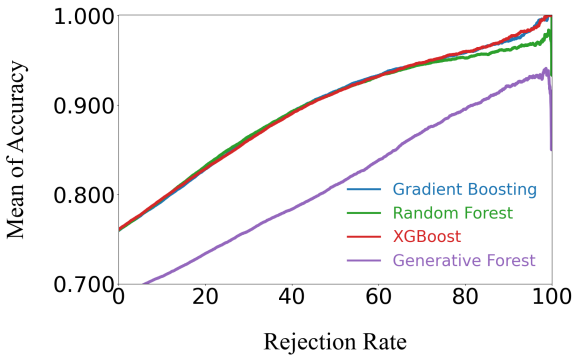
3.1.2 Comparison Between Models

Unlike r_{GeF}, our metric $r_{d_{COR}}$ is not restricted to Generative Forests, being applicable to any generative or discriminative model of interest. Hence, we can also make use of ARCs to verify if our robustness metric correlates with accuracy for different types of models. In our next experiment, we do this for Gradient Boosting (GB), Random Forests (RF), XGBoost [XGB, Chen and Guestrin, 2016] and GeFs.

Figure 3 presents the ARC of these four models in specific datasets. The leftmost value of the ARCs is the base accuracy of the models, indicating that Generative Forests was the least competitive model in both settings. For every model and both datasets, we observe that our robustness metric $r_{d_{COR}}$ still correlates with accuracy, but that the growth of the ARC varies based on dataset and model. In Figure 3a, the ARCs of the GB and of the XGB are similar at the start, being both superseded by the RF around a rejection rate of 20% (even though the RF started out with a lower base



(a) robot dataset (D_7)

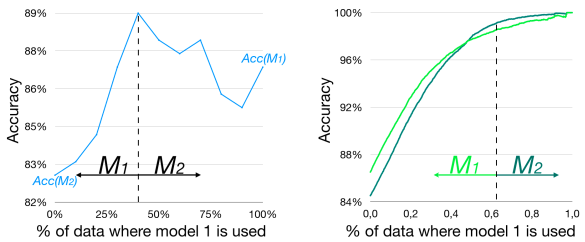


(b) students dataset (D_9)

Figure 3: Comparison between the ARCs of different models for two datasets when ordered by $r_{d_{COR}}$.

accuracy) and then exhibiting about the same behavior as RF around 40%, while the GeF only catches up around 70%. As for Figure 3b, the top three models initially perform equally well, with the GB and XGB eventually outperforming the other models around a rejection rate of 70%, while the GeF is again not competitive. These results not only imply that robustness correlates with accuracy, but also that different models might be the top performers at specific rejection rates. Moreover, the GeF underperformed in both cases, leading us to disregard it from further analyses.

One thing for which results like these can be used, is to gain an understanding of how well robustness is capable of assessing the reliability of the predictions of different models. Another, however, is to use them to increase performance in terms of accuracy. Most obviously, in a context where we can allow ourselves to reject a percentage of the data—for example because we can label those manually—we can use robustness-based ARCs to determine wich percentage to reject, and to determine which model performs best on the remaining data. However, it is also possible to use robustness to increase performance in contexts where rejection is not an option. This is what we come to next.



(a) RS-D (b) RS-I

Figure 4: Robustness-based strategies for DS. M_1 is the model with the best accuracy for the validation set, while M_2 is the second best.

3.2 DYNAMIC SELECTION OF CLASSIFIERS

To demonstrate the potential use of robustness metrics, we propose two robustness-based strategies (RS-D and RS-I) for Dynamic Selection (DS) of classifiers and compare them to other strategies in benchmark datasets. DS is an umbrella term for techniques that combine or choose between multiple base models, contingent on the set of features being used as input [Cruz et al., 2018]. Hence, depending on the values of the new sample, a DS classifier uses different base models as reference to provide a prediction.

Let M_1 and M_2 be the models with the highest and second highest accuracy in a hold-out validation set and let $r_1(\cdot)$ and $r_2(\cdot)$ be the their robustness metrics (since we focus solely on $r_{d_{\text{COR}}}$, we drop the subscript d_{COR} for notational convenience). For both DS classifier strategies, we start by ordering the validation set based on $r_2(\cdot)/r_1(\cdot)$. For instances that appear towards the beginning of the ordering, the prediction M_1 is more robust than that of M_2, and vice versa for instances towards the end of the ordering. The idea is now to only favor M_2 over M_1 when its robustness is considerably above that of M_1, so for instances towards the end of the ordering. Concretely, we choose a threshold t and, for any new instance x_{new}, we predict the corresponding class y_{new} using M_2 instead of M_1 only in cases when $r_2(x_{\text{new}})/r_1(x_{\text{new}}) > t$, and otherwise use M_1. The only difference between RS-D and RS-I is the way in which the threshold t is determined.

Figure 4 provides an illustration on how to derive the threshold for each strategy. On the horizontal axis of both graphs, we depict the percentage of the holdout validation data for which $r_2(x)/r_1(x)$ does not exceed the chosen t. The vertical axis depicts the accuracy on the same validation data, either for the DS classifier for this choice of t when applied to all validation data (Figure 4a) or for the individual models when applied to the data that exceeds t (Figure 4b). RS-D (Figure 4a) takes a Direct approach (hence the D). The idea is simply to choose t such that the resulting strategy maximizes the accuracy in the validation set. This is equivalent to finding the proportion p of the validation set with the lowest robustness difference that yields the highest accuracy

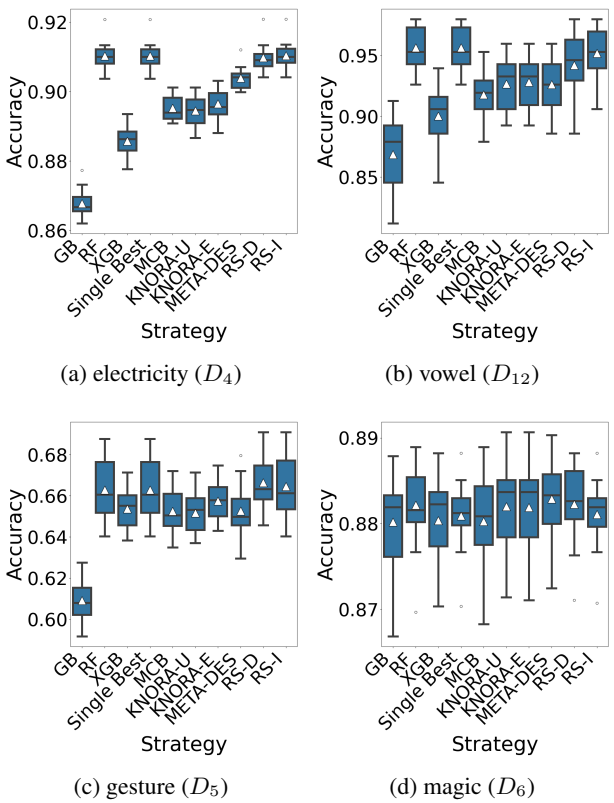
Table 2: Mean accuracy of each method for different datasets.
Best accuracy in bold, second best underlined.

Label	GB	RF	XGB	SB	MCB	KNORA-U	KNORA-E	META-DES	RS-D	RS-I
D_1	0.99742	0.99356	0.99388	**0.9971**	0.99678	0.99614	0.99678	0.99581	0.99549	0.99678
D_2	0.91598	0.91489	0.91594	0.91569	0.91568	0.9158	0.91573	**0.91657**	0.91594	0.91567
D_3	0.77126	0.76954	0.75517	0.76264	0.76494	0.76667	0.76667	0.76667	**0.76724**	0.76149
D_4	0.86772	0.91021	0.88572	0.91021	0.89522	0.89445	0.89647	0.90386	0.90975	**0.91033**
D_5	0.60909	0.66284	0.65367	0.66284	0.6525	0.6516	0.65744	0.65268	**0.6664**	0.66451
D_6	0.88019	0.88215	0.88036	0.88089	0.88024	0.88199	0.8819	**0.88288**	0.88227	0.88106
D_7	0.9974	0.99333	0.9965	0.99683	0.9956	0.99691	0.99683	0.99707	**0.9978**	0.99699
D_8	0.97406	0.97387	0.97483	0.97426	0.97426	0.97483	0.97483	0.97522	**0.97618**	0.97483
D_9	0.76581	0.76752	0.77023	0.77013	0.76842	0.77033	0.77033	**0.77193**	0.77073	0.77073
D_{10}	0.98079	0.9753	0.98103	0.98047	0.97942	0.982	0.98184	0.98208	**0.98257**	0.98184
D_{11}	0.97036	0.9721	0.97006	0.9709	0.97126	**0.9733**	0.97306	**0.9733**	0.97234	0.97174
D_{12}	0.86846	0.95615	0.90022	**0.95615**	0.91767	0.92662	0.92796	0.92617	0.94183	0.95168
D_{13}	0.85326	0.85175	0.85885	0.85442	0.85477	**0.85823**	0.85761	0.85761	0.85539	0.8561
D_{14}	0.85513	0.8553	0.86143	0.85841	0.85628	0.86019	**0.86045**	0.85957	0.8577	0.85965
D_{15}	0.94419	0.94651	0.94729	0.94496	0.94574	0.94496	0.94186	0.94186	0.94109	**0.94651**
Beats SB	5/15	4/15	7/15	-	4/15	10/15	9/15	10/15	10/15	11/15

when using M_1 to predict the class for this part of the data and M_2 to predict the rest. RS-I (Figure 4b), on the other hand, takes a more indirect approach. The idea here is to compare the ARCs of M_1 and M_2 for the ordering based on $r_2(\cdot)/r_1(\cdot)$ (so not based on $r_1(\cdot)$ and $r_2(\cdot)$, respectively), and to verify at what point (if at all) the ARC of M_2 has the largest accuracy advantage over M_1. So we choose t as the point for which, on the data for which $r_2(x)/r_1(x)$ exceeds t, M_2 provides the greatest accuracy gain compared to M_1.

Table 2 presents the mean accuracy of each DS technique for the datasets in Table 1. The base models used in the experiments were GB, RF and XGB. For each dataset, 15 different random train-validation-test splits were performed, with respective proportions of 0.7, 0.15 and 0.15. For each random split, the base models were fitted to the training set, with their hyperparameters being selected through grid search with a 5-fold cross-validation using the scikit-learn package [Pedregosa et al., 2011]. Then, the accuracy for the validation data was used for choosing the single best base model (SB) and for tuning the DS strategies. The competing DS methods (MCB, KNORA-U, KNORA-E and META-DES) can be found in Cruz et al. [2018] and are based on using KNN on the feature space to come up with the optimal model, using the validation set to decide the number of neighbors (maximum of 10). All the competing DS techniques were implemented through the DESlib library [Cruz et al., 2020].

Analysing the results in Table 2, we see that RS-D is the best performing strategy (indicated in bold) most often. Moreover, RS-D featured as one of the two best strategies in the majority of the datasets, along with META-DES. Furthermore, among the datasets in which RS-D wasn't listed as one of the two best strategies, RS-I was among the best in more than half of them. Compared to the Single Best (SB) strategy, RS-I was the alternative that most often had a better performance against it, even in circumstances where RS-D was not among the two best strategies. Moreover, in cases where one base model clearly outperforms the others—such as in the electricity (D_4) and vowel (D_{12}) datasets—all competing DS strategies based on the feature space underperform when compared to our robustness-based strategies. Figure 5 illustrates this situation in the two plots on the first row, while presenting more typical settings in the second row.



(a) electricity (D_4) (b) vowel (D_{12})

(c) gesture (D_5) (d) magic (D_6)

Figure 5: Boxplot of accuracies for each strategy and different datasets; white triangles represent the mean.

Table 3: Mean accuracy of each method under label corruption ($\rho = 0.05$).
Best accuracy in bold, second best underlined.

Label	GB	RF	XGB	SB	MCB	KNORA-U	KNORA-E	META-DES	RS-D	RS-I
D_1	0.98712	0.98583	0.99227	0.99259	0.99002	0.99163	0.9913	0.99195	0.99259	**0.99291**
D_2	0.91474	0.91421	0.91554	0.91501	0.9152	0.91579	0.91607	**0.91611**	0.91549	0.91514
D_3	0.75747	0.76954	0.75172	0.76092	0.75632	0.76264	0.76379	0.76092	**0.76552**	0.76149
D_4	0.86471	0.90057	0.88439	0.90057	0.88947	0.89033	0.89175	0.89528	0.90055	**0.90062**
D_5	0.59879	0.65807	0.64948	0.65601	0.64444	0.64534	0.652	0.64656	**0.66023**	0.65857
D_6	0.87461	0.88106	0.87655	0.87783	0.87748	0.87835	0.87767	0.87874	**0.87993**	0.87821
D_7	0.98926	0.96809	0.99373	**0.99373**	0.98689	0.99365	0.99365	**0.99373**	0.99292	**0.99373**
D_8	0.95677	0.91873	0.97003	0.96926	0.95543	0.96619	0.96619	0.96734	0.96849	**0.96945**
D_9	0.76361	0.76942	0.76882	0.76632	0.76642	0.76862	0.76872	0.76932	**0.77043**	0.76832
D_{10}	0.96578	0.87264	0.98136	0.98136	0.9632	0.977	0.97684	0.97571	0.98055	**0.98144**
D_{11}	0.96406	0.97138	0.96208	0.96952	0.9664	0.9697	0.96946	0.96982	**0.97024**	0.96982
D_{12}	0.85414	0.9387	0.90291	**0.93647**	0.90559	0.91902	0.91946	0.91678	0.92975	0.93512
D_{13}	0.8498	0.85362	0.8522	0.85051	0.85024	0.85424	**0.85619**	0.85317	0.85193	0.85078
D_{14}	0.8506	0.85353	0.85229	0.84953	0.85211	0.85451	0.85335	0.85442	**0.85486**	0.8506
D_{15}	0.94031	0.94496	0.94884	0.94806	0.94186	0.94729	0.94806	0.94729	**0.94884**	0.94729
Beats SB	1/15	8/15	6/15	-	3/15	7/15	5/15	6/15	9/15	12/15

3.2.1 Performance Under Label Corruption

While Table 2 shows that robustness can be used to build a competitive DS classifier, it only evaluates settings in which the test set comes from the same distribution as the rest of the data. To evaluate the performance of all strategies in a context of distribution shift, we perform label corruption on the training and validation sets, which works by uniformly changing the labels of the response variable for a proportion ρ of the data. This same technique has been previously explored in Li et al. [2020].

In Table 3, we set $\rho = 0.05$ and repeat the analyses. RS-D and RS-I remain as top performers in the new setting, but this time none of the competing DS strategies (not even META-DES) fares better than choosing the Single Best (SB) base model based on the accuracy in the validation set. Moreover, while RS-D still remains the highest performing strategy, RS-I is the technique that is most often superior to the Single Best. This pattern suggests that RS-I can be seen as the more reliable option, while RS-D is the one with greater chances of offering the best performance. Such behavior remains consistent even when higher levels of label corruption are applied, as demonstrated in Tables 4 and 5 in the supplementary material.

4 DISCUSSION

In this work, we have shown that robustness quantification can be brought to more general settings, being applicable to discriminative models and to settings with continuous features. Still, the choice of the dissimilarity function d of course remains somewhat arbitrary, justifying further studies into what other possible robustness metrics could have such properties.

Section 3.1.1 provides evidence—based on ARCs—that, even though our robustness metric does not take the architecture of the predictive model into account, it still outperforms the competing metric. This situation contrasts the findings of Detavernier and De Bock [2025a,b], which have shown that local robustness metrics can in fact perform better than global metrics. The reason for this discrepancy is still to be determined, with some possibilities being the fact that these works were limited to naive Bayes classifiers or that they were restricted to discrete features.

Considering the results in section 3.2, the use of robustness has presented superior performance in the context of Dynamic Selection of classifiers, especially when dealing with label corruption. One of the possible reasons is that standard DS techniques make assessments on the entirety of the feature space, which can lead to instability, whereas robustness offers a lower dimensional but rich representation of the data, leading to more stable procedures. Conversely, combining evaluations in the robustness space in a similar manner as that of other DS techniques could also offer improvements, being a potential topic for further research.

Besides robustness, there are other reliability metrics that could also offer a lower dimensional representation of the feature space, such as those used in uncertainty quantification [Hüllermeier and Waegeman, 2021], which could therefore be applied to the same Dynamic Selection tasks. More than arguing for the use of one over the other, we believe that each can bring different aspects of the model and the data to the spotlight, and should thus be tried in combination. In fact, combining robustness and uncertainty in a single metric has been shown to sometimes better correlate with accuracy than using each metric individually [Detavernier and De Bock, 2025b].

Lastly, since the strategies RS-D and RS-I are somewhat complementary of each other, devising a new technique that combines both of them is also a point of interest. Since this complementarity seems to be related to the presence or absence of a base model with superior predictive capacities than all others, one strategy to choose between using RS-D and RS-I would be to perform a hypothesis test. By

performing multiple train-validation splits and obtaining the accuracies of the models for each split, one could test if the mean accuracy between the best performing model and all others is significantly different. If so, this could corroborate using RS-I instead of RS-D.

5 PROOF OF THEOREM 1

Taking into account Equation (6), it clearly suffices to prove that $r_{d^*_{\mathrm{COR}}}(x) = r$, with $r = \frac{p(\hat{y}_1,x)}{p(\hat{y}_2,x)}$.

We first prove that $r_{d^*_{\mathrm{COR}}}(x) \leq r$, by constructing a measure $Q \in \mathbb{P}$ with density q such that $d^*_{\mathrm{COR}}(P,Q) = r$ and $q(\hat{y}_2|x) \geq q(\hat{y}_1|x)$.

Let

$$\pi_1 := P(Y = \hat{y}_1), \quad \pi_2 := P(Y = \hat{y}_2),$$
$$\lambda_- := \frac{\pi_1 + \pi_2}{\pi_1 + r\pi_2} \text{ and } \lambda_+ := \frac{r\pi_1 + r\pi_2}{\pi_1 + r\pi_2}$$

and consider the measurable function $L\colon \mathcal{Y} \times \mathcal{X} \to \mathbb{R}_{>0}$ defined as

$$L(y,x') := \begin{cases} \lambda_- & \text{if } y = \hat{y}_1, \\ \lambda_+ & \text{if } y = \hat{y}_2, \\ 1 & \text{otherwise.} \end{cases}$$

Then

$$\begin{aligned} \mathbb{E}_P[L] &= \lambda_+\pi_1 + \lambda_-\pi_2 + (1 - \pi_1 - \pi_2) \\ &= 1 + \pi_1(\lambda_+ - 1) + \pi_2(\lambda_- - 1) \\ &= 1 + \pi_1\frac{(r-1)\pi_2}{\pi_1 + r\pi_2} - \pi_2\frac{(r-1)\pi_1}{\pi_1 + r\pi_2} = 1. \end{aligned}$$

This implies that Q defined by $dQ/dP = L$ is a probability measure that is absolutely continuous w.r.t. P and has density $q(y,x') = L(y,x')p(y,x')$. Moreover, since $\lambda_- \leq 1 \leq \lambda_+$, we have $\operatorname{ess\,inf} L = \lambda_-$ and $\operatorname{ess\,sup} L = \lambda_+$, so it follows from Equation (5) that

$$d^*_{\mathrm{COR}}(P,Q) = \frac{\operatorname{ess\,sup} L}{\operatorname{ess\,inf} L} = \frac{\lambda_+}{\lambda_-} = r.$$

Finally,

$$\frac{q(\hat{y}_2|x)}{q(\hat{y}_1|x)} = \frac{q(\hat{y}_2,x)}{q(\hat{y}_1,x)} = \frac{\lambda_+ p(\hat{y}_2,x)}{\lambda_- p(\hat{y}_1,x)} = \frac{r}{r} = 1,$$

so $q(\hat{y}_2|x) \geq q(\hat{y}_1|x)$, which implies that the prediction $\hat{y}_1$ is not robust w.r.t. $\mathcal{P}_r$ for $d = d^*_{\mathrm{COR}}$ and thus that $r_{d^*_{\mathrm{COR}}}(x) \leq r$.

Next, we prove that $r_{d^*_{\mathrm{COR}}}(x) \geq r$. Let $Q \in \mathbb{P}$ be such that $d^*_{\mathrm{COR}}(P,Q) < r$, let q be its density and $L = q/p$ the likelihood ratio. Then $q(y,x) = L(y,x)p(y,x)$ and, due to Equation (5), $\operatorname{ess\,sup} L / \operatorname{ess\,inf} L < r$. For all $y \neq \hat{y}_1$, since $p(y,x) \leq p(\hat{y}_2,x)$, this implies that

$$\begin{aligned} \frac{q(\hat{y}_1|x)}{q(y|x)} &= \frac{q(\hat{y}_1,x)}{q(y,x)} = \frac{p(\hat{y}_1,x)}{p(y,x)}\frac{L(\hat{y}_1,x)}{L(y,x)} \\ &\geq \frac{p(\hat{y}_1,x)}{p(y,x)}\frac{\operatorname{ess\,inf} L}{\operatorname{ess\,sup} L} > \frac{p(\hat{y}_1,x)}{p(y,x)}\frac{1}{r} \\ &\geq \frac{p(\hat{y}_1,x)}{p(\hat{y}_2,x)}\frac{1}{r} = 1. \end{aligned}$$

So $q(\hat{y}_1|x) > q(y|x)$ for all $y \neq \hat{y}_1$, which means that Q predicts the same class $\hat{y}_1$ as P. Since this is true for every $Q \in \mathbb{P}$ such that $d^*_{\mathrm{COR}}(P,Q) < r$, we find that the prediction $\hat{y}_1$ is robust w.r.t. $\mathcal{P}_\delta$ for all $\delta < r$ and therefore that $r_{d^*_{\mathrm{COR}}}(x) \geq r$.

Author Contributions

The authors contributed equally to the development of the ideas and the writing of the paper. J. De Bock did the proof of Theorem 1 and R. F. L. Lassance created the code and ran the experiments.

Acknowledgements

R. F. L. Lassance: this work was carried out with the support of the Coordination for the Improvement of Higher Education Personnel – Brazil (CAPES) – Financing Code 001. J. De Bock: a BOF basic research funding grant of Ghent University was provided in the period that this research was carried out. The authors thank Adrián Detavernier and Rafael B. Stern for fruitful conversations and suggestions on how to make the contributions of robustness quantification more meaningful.

References

Thomas Augustin, Frank P. A. Coolen, Gert de Cooman, and Matthias C. M. Troffaes, editors. *Introduction to imprecise probabilities*. Wiley Series in Probability and Statistics. Wiley-Blackwell, Hoboken, NJ, May 2014. doi: 10.1002/9781118763117.

Leo Breiman. Random forests. *Mach. Learn.*, 45(1): 5–32, October 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324.

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*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939785.

Alvaro H. C. Correia and Cassio P. de Campos. Towards scalable and robust sum-product networks. In Nahla Ben Amor, Benjamin Quost, and Martin Theobald, editors, *Scalable Uncertainty Management*, pages 409–422, Cham, 2019. Springer International Publishing. doi: 10.1007/978-3-030-35514-2_31.

Alvaro H. C. Correia, Robert Peharz, and Cassio de Campos. Towards robust classification with deep generative forests, 2020. URL https://arxiv.org/abs/2007.05721.

Rafael M. O. Cruz, Luiz G. Hafemann, Robert Sabourin, and George D. C. Cavalcanti. Deslib: A dynamic ensemble selection library in python. *Journal of Machine Learning Research*, 21(8):1–5, 2020. URL http://jmlr.org/papers/v21/18-144.html.

Rafael M.O. Cruz, Robert Sabourin, and George D.C. Cavalcanti. Dynamic classifier selection: Recent advances and perspectives. *Information Fusion*, 41:195–216, 2018. ISSN 1566-2535. doi: 10.1016/j.inffus.2017.09.010.

Jasper De Bock, Cassio P. de Campos, and Alessandro Antonucci. Global sensitivity analysis for map inference in graphical models. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/f9d3c99bd6cbf2d694266e7760ee1ed6-Paper.pdf.

Adrián Detavernier and Jasper De Bock. Robustness quantification : a new method for assessing the reliability of the predictions of a classifier. In *Proceedings of the Fourteenth International Symposium on Imprecise Probabilities: Theories and Applications*, volume 290, pages 126–136, 2025a. URL https://proceedings.mlr.press/v290/detavernier25a.html.

Adrián Detavernier and Jasper De Bock. Robustness and uncertainty: two complementary aspects of the reliability of the predictions of a classifier, 2025b. URL https://arxiv.org/abs/2512.15492.

Lutz Dümbgen, Richard J. Samworth, and Jon A. Wellner. Bounding distributional errors via density ratios. *Bernoulli*, 27(2), May 2021. ISSN 1350-7265. doi: 10.3150/20-bej1256.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*, 110(3):457–506, Mar 2021. ISSN 1573-0565. doi: 10.1007/s10994-021-05946-3.

Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. UCI machine learning repository, 2023. URL https://archive.ics.uci.edu.

Meizhu Li, Shaoguang Huang, Jasper De Bock, Gert de Cooman, and Aleksandra Pižurica. A robust dynamic classifier selection approach for hyperspectral images with imprecise label information. *Sensors*, 20(18), 2020. ISSN 1424-8220. doi: 10.3390/s20185262.

Christoph Molnar. *Interpretable Machine Learning*. https://christophm.github.io/interpretable-ml-book/, 3 edition, 2025. ISBN 978-3-911578-03-5. URL https://christophm.github.io/interpretable-ml-book.

Ignacio Montes, Enrique Miranda, and Sébastien Destercke. Unifying neighbourhood and distortion models: part i – new results on old models. *International Journal of General Systems*, 49(6):602–635, 2020. doi: 10.1080/03081079.2020.1778682.

Awais Muhammad and Sung-Ho Bae. A survey on efficient methods for adversarial robustness. *IEEE Access*, 10:118815–118830, 2022. doi: 10.1109/ACCESS.2022.3216291.

Malik Sajjad Ahmed Nadeem, Jean-Daniel Zucker, and Blaise Hanczar. Accuracy-rejection curves (arcs) for comparing classification methods with a reject option. In Sašo Džeroski, Pierre Guerts, and Juho Rousu, editors, *Proceedings of the third International Workshop on Machine Learning in Systems Biology*, volume 8 of *Proceedings of Machine Learning Research*, pages 65–81, Ljubljana, Slovenia, 05–06 Sep 2009. PMLR. URL https://proceedings.mlr.press/v8/nadeem10a.html.

Cathy O'Neil. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Crown Publishing Group, USA, 2016. ISBN 0553418815.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12 (85):2825–2830, 2011. URL http://jmlr.org/papers/v12/pedregosa11a.html.

Joseph D. Romano, Trang T. Le, William La Cava, John T. Gregg, Daniel J. Goldberg, Praneel Chakraborty, Natasha L. Ray, Daniel Himmelstein, Weixuan Fu, and Jason H. Moore. PMLB v1.0: an open source dataset collection for benchmarking machine learning methods. *arXiv preprint arXiv:2012.00058v2*, 2021. URL https://arxiv.org/abs/2012.00058.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5): 206–215, May 2019. ISSN 2522-5839. doi: 10.1038/s42256-019-0048-x.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. OpenML: networked science in machine learning. *SIGKDD Explor. Newsl.*, 15(2):49–60, June 2014. ISSN 1931-0145. doi: 10.1145/2641190.2641198.

Peter Walley. *Statistical Reasoning with Imprecise Probabilities*, volume 42 of *Monographs on Statistics and Applied Probability*. Chapman and Hall, London, 1991.

Robustness Quantification for Discriminative Models (Supplementary Material)

Rodrigo F. L. Lassance[1,2,3] **Jasper De Bock**[1]

[1]Foundations Lab for imprecise probabilities (FLip), Ghent University, Ghent, Belgium
[2]Statistics Dept., Federal University of São Carlos, São Carlos, São Paulo, Brazil
[3]Institute of Mathematic and Computer Sciences, University of São Paulo, São Carlos, São Paulo, Brazil

Table 4: Mean accuracy of each method under label corruption ($\rho = 0.1$).
Best accuracy in bold, second best underlined.

Label	GB	RF	XGB	SB	MCB	KNORA-U	KNORA-E	META-DES	RS-D	RS-I
D_1	0.98068	0.98261	0.98647	0.98841	0.98583	0.98937	0.98969	**0.99002**	0.98776	0.98969
D_2	0.91351	0.90522	0.91446	0.91385	0.91166	0.91145	0.9113	0.91284	0.91423	**0.91429**
D_3	0.74713	0.76954	0.7546	0.75402	**0.76609**	0.76149	0.7569	0.76207	0.76379	0.75575
D_4	0.85802	0.89204	0.87812	0.89204	0.88107	0.88344	0.88445	0.88689	**0.89383**	0.89219
D_5	0.58817	0.65092	0.63927	0.64975	0.63437	0.6305	0.64076	0.63711	**0.65245**	0.65074
D_6	0.87064	0.87786	0.87342	0.87648	0.87416	0.87489	0.8747	0.87564	**0.87741**	0.87687
D_7	0.98803	0.9676	0.98958	0.98942	0.98551	0.99088	0.99105	**0.99162**	0.99129	0.99064
D_8	0.94409	0.92065	0.96119	0.96061	0.95293	0.95793	0.95812	0.95869	0.95985	**0.96081**
D_9	0.75669	0.76622	0.7612	0.7614	0.76221	0.76241	0.76371	0.76411	**0.76571**	0.76361
D_{10}	0.95956	0.86772	0.97869	**0.97869**	0.96053	0.97103	0.97256	0.97119	0.97635	0.97797
D_{11}	0.9613	0.96874	0.95842	**0.9679**	0.96202	0.96754	0.9673	0.96748	0.96778	0.9676
D_{12}	0.82685	0.91991	0.87919	**0.92081**	0.8868	0.89306	0.89575	0.89083	0.90917	0.9132
D_{13}	0.84412	0.85237	0.84874	0.84891	0.85166	0.85184	**0.85344**	0.85122	0.849	0.84971
D_{14}	0.84669	0.85211	0.85069	0.84927	0.84927	0.85149	0.85246	0.85477	**0.85584**	0.85078
D_{15}	0.92248	0.94574	0.93721	0.94186	0.94109	0.94186	0.94419	0.93876	**0.94496**	0.94264
Beats SB	0/15	8/15	6/15	-	3/15	6/15	7/15	6/15	10/15	12/15

Table 5: Mean accuracy of each method under label corruption ($\rho = 0.2$).
Best accuracy in bold, second best underlined.

Label	GB	RF	XGB	SB	MCB	KNORA-U	KNORA-E	META-DES	RS-D	RS-I
D_1	0.94847	0.97359	0.97262	0.97327	0.96844	0.97874	**0.97939**	0.97617	0.97649	0.9752
D_2	0.91295	0.90548	0.91228	0.91313	0.91017	0.9105	0.91051	0.91242	**0.91338**	0.91314
D_3	0.71322	0.75287	0.73908	0.74195	0.73046	**0.74483**	0.74425	0.7431	0.73851	0.73851
D_4	0.84607	0.85835	0.8595	0.85955	0.859	0.86329	0.86388	0.86241	**0.86998**	0.86176
D_5	0.56644	0.64197	0.62231	0.63941	0.61831	0.61165	0.62434	0.62092	**0.64462**	0.64017
D_6	0.86403	0.8665	0.86578	0.86529	0.86508	0.86802	0.86847	**0.8687**	0.86858	0.86559
D_7	0.98266	0.96443	0.9825	0.98063	0.98128	0.98584	0.98689	**0.98746**	0.98665	0.98396
D_8	0.92277	0.92027	0.9414	0.93929	0.93372	0.94159	**0.94467**	0.94428	0.94409	0.94121
D_9	0.75028	0.75228	0.74657	0.74937	0.74817	0.75388	**0.75589**	0.75409	0.75519	0.75258
D_{10}	0.94649	0.86489	0.96594	0.96594	0.94496	0.95642	0.96287	0.95932	0.96594	**0.96626**
D_{11}	0.95014	0.96538	0.94287	**0.96538**	0.95476	0.95884	0.9586	0.9589	0.96244	0.96436
D_{12}	0.74989	0.8783	0.82998	**0.8783**	0.83669	0.83982	0.84743	0.85101	0.87293	0.8774
D_{13}	0.83604	0.85308	0.83356	0.8474	0.8419	0.84359	0.84634	0.84634	0.84758	**0.84838**
D_{14}	0.83595	0.85362	0.83693	0.85175	0.8435	0.8466	0.8498	0.849	**0.85255**	0.85166
D_{15}	0.88915	0.92946	0.90775	0.92016	0.9093	0.92481	0.92558	0.92403	**0.92946**	0.92403
Beats SB	2/15	8/15	5/15	-	1/15	8/15	9/15	8/15	11/15	11/15