DOCTOR: A Simple Method for Detecting Misclassification Errors

Federica Granese, Marco Romanelli, Daniele Gorla, Catuscia Palamidessi, Pablo Piantanida

DOCTOR: A Simple Method for Detecting Misclassification Errors

Federica Granese[*,†]
Lix, Inria, Institute Polytechnique de Paris,
Sapienza University of Rome
federica.granese@inria.fr

Marco Romanelli[*]
L2S, CentraleSupélec,
CNRS, Université Paris Saclay
marco.romanelli@centralesupelec.fr

Daniele Gorla
Sapienza University of Rome
gorla@di.uniroma1.it

Catuscia Palamidessi[†]
Lix, Inria, Institute Polytechnique de Paris,
catuscia@lix.polytechnique.fr

Pablo Piantanida[‡]
L2S, CentraleSupélec,
CNRS, Université Paris Saclay
pablo.piantanida@centralesupelec.fr

Abstract

Deep neural networks (DNNs) have shown to perform very well on large scale object recognition problems and lead to widespread use for real-world applications, including situations where DNN are implemented as "black boxes". A promising approach to secure their use is to accept decisions that are likely to be correct while discarding the others. In this work, we propose DOCTOR, a simple method that aims to identify whether the prediction of a DNN classifier should (or should not) be trusted so that, consequently, it would be possible to accept it or to reject it. Two scenarios are investigated: Totally Black Box (TBB) where only the soft-predictions are available and Partially Black Box (PBB) where gradient-propagation to perform input pre-processing is allowed. Empirically, we show that DOCTOR outperforms all state-of-the-art methods on various well-known images and sentiment analysis datasets. In particular, we observe a reduction of up to 4% of the false rejection rate (FRR) in the PBB scenario. DOCTOR can be applied to any pre-trained model, it does not require prior information about the underlying dataset and is as simple as the simplest available methods in the literature.

1 Introduction

With the advancement of state-of-the-art Deep Neural Networks (DNNs), there has been rapid adoption of these technologies in a broad range of applications to critical systems, such as autonomous driving vehicles or industrial robots, including–but not limited to–classification and decision making tasks. Nevertheless, these solutions still exhibit unwanted behaviors as they tend to be overconfident even in presence of wrong decisions [17]. Developing methods and tools to make these algorithms

[*]These authors contributed equally to this work.

[†]This paper is supported by the ERC project Hypatia under the European Unions Horizon 2020 research and innovation program. Grant agreement N. 835294.

[‡]This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement N. 792464.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

reliable, in particular for non-specialists who may treat them as "black boxes" with no further checks, constitutes a core challenge. Recently, the study of safety AI methods has gained ground, and many efforts have been made in several areas [7–9, 11, 26, 35, 36]. In this paper, we investigate a simple method capable of detecting whether a prediction of a classifier is likely to be correct, and therefore should be trusted, or it is not, and should be rejected.

Deep learning pursues the idea of learning effective representations from the data itself by training with the implicit assumption that the test data distribution should be similar to the training data distribution. However, when applied to real-world tasks, this assumption does not hold true, leading to a significant increase of misclassification errors. Although classic approaches to Out-Of-Distribution (OOD) detection [1, 12, 21, 23, 33] are not directly concerned with detecting misclassification errors, they are intended to prevent those errors indirectly by identifying potential drifts of the testing distribution. What the above ODD methods have in common with our work is that samples drawn from the in-distribution are more likely to be correctly classified than those from a different distribution. Indeed, the model's soft-predictions for in-distribution samples tend to be generally peaky in correspondence to the correct class label while they tend to be less peaky for input samples drawn from a different distribution [12]. In general, most of these works consider *white-box* scenarios, where the hidden layers of the architecture are accessible or the corresponding weights are tuned during the training phase. A very effective approach to OOD detection is ODIN [23] which involves the use of temperature scaling and the addition of small perturbations to input samples. A related solution is introduced in [6] where the maximum soft-probability is called *softmax response*. Within this approach, the softmax response decides whether the classifier is confident enough in its prediction or not. A different approach to OOD detection is given by the use of the Mahalanobis distance [14, 22], which consists in calculating how much the observed out-distribution sample deviate from the in-distribution ones but assuming the latter are given.

1.1 Summary of contributions

Our work tackles the problem of identifying whether the prediction of a classifier should or should not be trusted, no matter if they are made on out or in-distribution samples, and advances the state-of-the-art in multiple ways.

- From the theoretical point of view, we derive the trade-off between two types of error probabilities: Type-I, that refers to the rejection of the classification for an input that would be correctly classified, and Type-II, that refers to the acceptance of the classification for an input that would not be correctly classified (Proposition 3.1). The characterization of the optimal discriminator in eq. (10) allows us to devise a feasible implementation of it, based on the softmax probability (Proposition 3.2).
- From the algorithmic point of view, inspired by our theoretical analysis, we propose DOCTOR a new discriminator (Definition 2), which yields a simple and flexible framework to detect whether a decision made by a model is likely to be correct or not. We distinguish two scenarios under which DOCTOR can be deployed: Totally Black Box (TBB) where only the soft-predictions are available, hence gradient-propagation to perform input pre-processing is not allowed, and Partially Black Box (PBB) where we further allow method-specific inputs perturbations.
- From the experimental point of view, we show that DOCTOR outperforms comparable state-of-the-art methods (e.g., ODIN [23], softmax response [6] and Mahalanobis distance [22]) on datasets including both in-distribution and out-of-distribution samples, and different architectures with various expressibilities, under both TBB and PBB. A key ingredient of DOCTOR is to fully exploit all available information contained in the soft-probabilities of the predictions (not only their maximum).

1.2 Related works

Recent works have shown that the accuracy of a classifier and its ability to output soft-predictions that represent the true posteriors estimate can be totally disjointed [9, 19, 20]. Furthermore, models often tend to be overconfident about their decision even when their predictions fail [11, 17]. This motivates a novel research area that strives to develop methods to assess when decisions made by classifiers should or should not be trusted. Although the detection of OOD samples is a different (domain)

problem, it is naturally expected that samples from a distribution that is significantly different from the training one cannot be correctly classified. In [23], the authors propose a method which increases the peakiness of the softmax output by perturbing the input samples and applying temperature scaling [9, 13, 29] to the classifier logits in order to better detect in-distribution samples. It is worth noticing that this method requires additional information on the internal structure of the latent code of the model. A very different approach [14, 22] tackles OOD detection by using the *Mahalanobis distance*. Although this approach appears to be more powerful, it also requires additional samples to learn the mean by class and the covariance matrix of the in-distribution. In [4], classifiers are trained to output calibrated confidence estimates that are used to perform OOD detection. A related line of research is concerned with the problem of *selective predictions* (aka *reject options*) in deep neural networks. The main motivation for selective prediction is reducing the error rate by abstaining from prediction when in doubt, while keeping the number of correctly classified samples as high as possible [5–7]. The idea is to combine classifiers with *rejection functions* by observing the classifiers' output, without using any supervision, to decide whether to accept or to reject the classification outcome. In [6], the authors introduce *softmax response*, a rejection function which compares the maximum soft-probability to a pre-determined threshold to decide whether to accept or reject the class prediction given by the model.

2 Main Definitions and Preliminaries

2.1 Basic definitions

We start by introducing some definitions and background; then, we describe our statistical model and some useful properties about the underlying detection problem. Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the (possibly continuous) feature space and let $\mathcal{Y} = \{1, \dots, C\}$ denote the concept of the label space related to some task of interest. Moreover, let p_{XY} be the underlying (unknown) probability density function (pdf) over $\mathcal{X} \times \mathcal{Y}$. Let $\mathcal{D}_n = \big\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\big\} \sim p_{XY}$ be a random realization of n i.i.d. samples according to p_{XY} denoting the *training set*, where $\mathbf{x}_i \in \mathcal{X}$ is the input (feature), $y_i \in \mathcal{Y}$ is the output class among C possible classes and n denotes the size of the training set. A predictor $f_{\mathcal{D}_n} : \mathcal{X} \to \mathcal{Y}$ uses the inferred model $P_{\widehat{Y}|X} \equiv P_{\widehat{Y}|X}(y|\mathbf{x}; \mathcal{D}_n)$ based on the training set,

$$f_{\mathcal{D}_n}(\mathbf{x}) \equiv f_n(\mathbf{x}; \mathcal{D}_n) \triangleq \arg\max_{y \in \mathcal{Y}} P_{\widehat{Y}|X}(y|\mathbf{x}; \mathcal{D}_n),$$

and tries to approximate the optimal (Bayes) decision rule $f^\star(\mathbf{x}) \triangleq \arg\max_{y \in \mathcal{Y}} P_{Y|X}(y|\mathbf{x})$. Notice that $P_{\widehat{Y}|X}$ can be interpreted as the prediction of the class (label) posterior probability given a sample (e.g., $P_{\widehat{Y}|X}(y\,|\mathbf{x}) \equiv \text{softmax}(\mathbf{x})_y$), while $P_{Y|X}$ is the true (unknown) probability. In several practical scenarios $P_{\widehat{Y}|X}$ does not perfectly match $P_{Y|X}$ and still $f_{\mathcal{D}_n} \approx f^\star$ (cf. [9]).

2.2 Error variable

Let $E(\mathbf{x}) \triangleq \mathbb{1}\left[Y \neq f_{\mathcal{D}_n}(\mathbf{x})\right]$ denote the error variable for a given $\mathbf{x} \in \mathcal{X}$ corresponding to $f_{\mathcal{D}_n}$, i.e., where we denote with $\mathbb{1}[\mathcal{E}]$ the indicator vector which outputs 1 if the event $\mathcal{E}$ is true and 0 otherwise. Similarly, we can define the self-error variable $\widehat{E}(\mathbf{x}) \triangleq \mathbb{1}\big[\widehat{Y} \neq f_{\mathcal{D}_n}(\mathbf{x})\big]$ also corresponding to the inferred predictor $f_{\mathcal{D}_n}$ but based on the prediction model $P_{\widehat{Y}|X}$ of the class posterior probability. Notice that $\widehat{E}(\mathbf{x})$ is observable since the underlying distribution is known. However, $E(\mathbf{x})$ cannot be observed and in general these binary variables do not coincide.

At this stage, it is convenient to introduce the notions of *probability of classification error* for a given $\mathbf{x} \in \mathcal{X}$ w.r.t. both the true class posterior and the predicted probabilities:

$$\text{Pe}(\mathbf{x}) \triangleq \mathbb{E}\left[E(\mathbf{x})|\mathbf{x}\right] = 1 - P_{Y|X}\left(f_{\mathcal{D}_n}(\mathbf{x})|\mathbf{x}\right), \tag{1}$$

$$\widehat{\text{Pe}}(\mathbf{x}) \triangleq \mathbb{E}\left[\widehat{E}(\mathbf{x})|\mathbf{x}\right] = 1 - P_{\widehat{Y}|X}\left(f_{\mathcal{D}_n}(\mathbf{x})|\mathbf{x}\right). \tag{2}$$

Notice that $\widehat{\text{Pe}}(\mathbf{x})$ represents the probability of misclassification of the sample $\mathbf{x}$ with respect to the softmax probability $P_{\widehat{Y}|X}$, which can be interpreted as the model's approximation of nature $P_{Y|X}$. Such approximation is close when the model is well-calibrated. Obviously, $\text{Pe}^\star(\mathbf{x}) \leq \text{Pe}(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{X}$, where $\text{Pe}^\star(\mathbf{x})$ corresponds to the minimum error of the Bayes classifier:

$\text{Pe}^\star(\mathbf{x}) = 1 - P_{Y|X}\left(f^\star(\mathbf{x})|\mathbf{x}\right)$. It is worth mentioning that, by averaging (1) over the data distribution, we obtain the error rate of the classifier $f_{\mathcal{D}_n}$. Although $\widehat{\text{Pe}}(\mathbf{x})$ provides a valuable candidate to infer the unknown error variable $E(\mathbf{x})$, it is easy to check that

$$\max\left\{\text{Pe}(\mathbf{x}), \widehat{\text{Pe}}(\mathbf{x})\right\} - \Pr\left(\widehat{Y} = Y|\mathbf{x}\right) \leq \Pr\left\{\widehat{E}(\mathbf{x}) \neq E(\mathbf{x})|\mathbf{x}\right\} \leq \Pr\left(\widehat{Y} \neq Y|\mathbf{x}\right), \tag{3}$$

which in particular implies that the error incurred in using $\widehat{E}(\mathbf{x})$ to predict $E(\mathbf{x})$ is lower bounded by the classification error per sample (1). The proofs are in Supplementary material (Appendix A.3).

In this paper, we aim at identifying a discriminator capable of distinguishing between inputs $\mathbf{x}$ for which we can trust the predictions of the classifier $f_{\mathcal{D}_n}(\mathbf{x})$ (i.e., $E(\mathbf{x}) = 0$) and those for which we should not trust predictions (i.e., $E(\mathbf{x}) = 1$). In the next section, we will show that the function $\text{Pe}(\mathbf{x}) : \mathcal{X} \mapsto [0, 1]$ plays a central role in the characterization of the optimal discriminator. However, $\text{Pe}(\mathbf{x})$ is not available in practical scenarios and the direct estimation (e.g., based on pairs of inputs and labels) of the true class posterior probability $P_{Y|X}$ cannot be performed. Notice that it is not possible to sample the conditional pdf $P_{Y|X}$ for each input $\mathbf{x} \in \mathcal{X}$. As a matter of fact, it is well-known that the application of direct methods for this estimation will lead to ill-posed problems, as shown in [32].

2.3 Statistical model for detection

Given a data sample $\mathbf{x} \in \mathcal{X}$ and an unobserved random label $y \in \mathcal{Y}$ drawn from the unknown distribution p_{XY}, we wish to predict the realization of the unobserved error variable $E \triangleq \mathbb{1}[Y \neq f_{\mathcal{D}_n}(\mathbf{X})]$. To this end, we will model the data distribution as a mixture pdfs,

$$p_{XY}(\mathbf{x}, y) \equiv P_E(1)p_{XY|E}(\mathbf{x}, y|1) + P_E(0)p_{XY|E}(\mathbf{x}, y|0),$$

where $p_{XY|E}(\mathbf{x}, y|1)$ denotes the pdf truncated to the error event $\{E = 1\}$ (i.e., the hard decision fails) and $p_{XY|E}(\mathbf{x}, y|0)$ is the pdf truncated to the success event $\{E = 0\}$ (i.e., the hard decision succeeds). By taking the marginal of p_{XY} over the labels, we obtain: $p_X(\mathbf{x}) = P_E(1)p_{X|E}(\mathbf{x}|1) + P_E(0)p_{X|E}(\mathbf{x}|0)$. First, observe that the problem at hand is to infer E from $(\mathbf{x}, P_{\widehat{Y}|X})$ since Y is not observed. Second, we further emphasize that in the present framework we assume that there are no available (extra) samples for training a discriminator to distinguish between $p_{X|E}(\mathbf{x}|0)$ and $p_{X|E}(\mathbf{x}|1)$. It is worth mentioning that a well-trained classifier would imply $P_E(1) \ll P_E(0)$, since in that case we should have very few classification errors. However, this also implies that it would be very unlikely to have enough samples available to train a good enough discriminator.

3 Performance Metrics and Discriminators

3.1 Performance metrics and optimal discriminator

We aim to distinguish between samples for which the predictions cannot be trusted and samples for which predictions should be trusted. We first state the optimal rejection region, given by (4), where we suppose the existence of an oracle who knows all the involved probability distributions.

Definition 1 (Most powerful discriminator)**.** For any $0 < \gamma < \infty$, define the decision region:

$$\mathcal{A}(\gamma) \triangleq \left\{\mathbf{x} \in \mathcal{X} : p_{X|E}(\mathbf{x}|1) > \gamma \cdot p_{X|E}(\mathbf{x}|0)\right\}. \tag{4}$$

The most powerful (Oracle) discriminator at threshold γ is defined by setting $D(\mathbf{x}, \gamma) = 1$ for all $\mathbf{x} \in \mathcal{A}(\gamma)$ for which the prediction is rejected (i.e., $\widehat{E} = 1$) and otherwise $D(\mathbf{x}, \gamma) = 0$ for all $\mathbf{x} \notin \mathcal{A}(\gamma)$ for which the prediction is accepted.

In Proposition 3.1, we establish the characterization of the fundamental performance of the most powerful (Oracle) discriminator by providing a lower bound on the error achieved by any discriminator and show that this bound is achievable by setting $\gamma = 1$. Furthermore, we connect this result to the Bayesian error rate of this optimal discriminator.

Proposition 3.1 (Performance of the discriminator)**.** For any given decision region $\mathcal{A} \subset \mathcal{X}$, let

$$\epsilon_0(\mathcal{A}) \triangleq \int_{\mathcal{A}} p_{X|E}(\mathbf{x}|0)d\mathbf{x}, \quad \text{and} \quad \epsilon_1(\mathcal{A}^c) \triangleq \int_{\mathcal{A}^c} p_{X|E}(\mathbf{x}|1)d\mathbf{x}\,, \tag{5}$$

be the Type-I (rejection of the class prediction of an input $\mathbf{x}$ that would be correctly classified) and Type-II (acceptance of the class prediction of an input $\mathbf{x}$ that would not be correctly classified) error probability, respectively. Then,

$$\epsilon_0(\mathcal{A}) + \epsilon_1(\mathcal{A}^c) \geq 1 - \left\| p_{X|E=1} - p_{X|E=0} \right\|_{\mathrm{TV}} \tag{6}$$

$$= 1 - \frac{1}{2} \int_{\mathcal{X}} |p_{X|E=1}(\mathbf{x}) - p_{X|E=0}(\mathbf{x})| d\mathbf{x}. \tag{7}$$

Equality is achieved by choosing the optimal decision region $\mathcal{A}^\star \equiv \mathcal{A}(1)$ in Definition 1. If the hypotheses are equally distributed, the minimum Bayesian error satisfies:

$$2 \Pr\{D(\mathbf{X}) \neq E(\mathbf{X})\} \geq 1 - \left\| p_{X|E=1} - p_{X|E=0} \right\|_{\mathrm{TV}}. \tag{8}$$

Equality is achieved by using the optimal decision region.

Expressions (7) and (8) provide lower bounds for the total error of an arbitrary discriminator. The proof of this proposition is relegated to the Supplementary material (Appendix A.1). Using Bayes we can rewrite (4) via the posteriors as:

$$\mathcal{A}(\gamma) = \left\{ \mathbf{x} \in \mathcal{X} : P_{E|X}(1|\mathbf{x}) P_E(0) > \gamma \cdot \left(1 - P_{E|X}(1|\mathbf{x})\right) P_E(1) \right\}. \tag{9}$$

From (9), it is easy to check that $P_{E|X}(1|\mathbf{x}) = 1 - P_{Y|X}\left(f_{\mathcal{D}_n}(\mathbf{x})|\mathbf{x}\right) = \mathrm{Pe}(\mathbf{x})$, and hence, the decision region $\mathcal{A}(\gamma)$ can be reformulated as:

$$\mathcal{A}(\gamma') = \left\{ \mathbf{x} \in \mathcal{X} : \frac{\mathrm{Pe}(\mathbf{x})}{1 - \mathrm{Pe}(\mathbf{x})} > \gamma' \right\} = \left\{ \mathbf{x} \in \mathcal{X} : \mathrm{Pe}(\mathbf{x}) > \frac{\gamma'}{(\gamma' + 1)} \right\}, \tag{10}$$

where $\gamma' \triangleq \gamma \cdot \frac{P_E(1)}{P_E(0)}$ and $0 < \gamma' < \infty$. According to (10) and Proposition (3.1), the optimal discriminator is given by $D^\star(\mathbf{x}, \gamma') = 1$, whenever $\mathbf{x} \in \mathcal{A}(\gamma')$, and $D^\star(\mathbf{x}, \gamma') = 0$, otherwise. The main difficulty arises here since the error probability function of an input: $\mathbf{x} \mapsto \mathrm{Pe}(\mathbf{x})$ is not known and in general cannot be learned from training samples.

3.2 DOCTOR discriminator

We start by deriving an approximation to the unknown function $\mathbf{x} \mapsto \mathrm{Pe}(\mathbf{x})$ which can be used to devise the decision region in expression (10). First, we state the following:

Proposition 3.2. Let $\widehat{\mathrm{g}}(\mathbf{x})$ be defined by

$$1 - \widehat{\mathrm{g}}(\mathbf{x}) \triangleq \sum_{y \in \mathcal{Y}} P_{\widehat{Y}|X}(y|\mathbf{x}) \Pr\left(\widehat{Y} \neq y|\mathbf{x}\right) = 1 - \sum_{y \in \mathcal{Y}} P^2_{\widehat{Y}|X}(y|\mathbf{x}), \tag{11}$$

for each $\mathbf{x} \in \mathcal{X}$, which indicates the probability of incorrectly classifying a feature $\mathbf{x}$ if it was randomly labeled according to the model distribution $P_{\widehat{Y}|X}$ trained based on the dataset. Then,

$$(1 - \sqrt{\widehat{\mathrm{g}}(\mathbf{x})}) - \Delta(\mathbf{x}) \leq \mathrm{Pe}(\mathbf{x}) \leq (1 - \widehat{\mathrm{g}}(\mathbf{x})) + \Delta(\mathbf{x}), \tag{12}$$

where $\Delta(\mathbf{x}) \triangleq 2\sqrt{2\,\mathrm{KL}\big(P_{Y|X}(\cdot|\mathbf{x}) \| P_{\widehat{Y}|X}(\cdot|\mathbf{x})\big)}$ and denotes the Kullback–Leibler (KL) divergence (further details are provided in Supplementary material Appendix A.2).

3.3 Discussion

It is worth emphasizing that expressions in (12) provide bounds to the unknown function $\mathbf{x} \mapsto \mathrm{Pe}(\mathbf{x})$ using a known statistics $\mathbf{x} \mapsto 1 - \widehat{\mathrm{g}}(\mathbf{x})$, which is based on the soft-probability of the predictor. On the other hand, $0 \leq \widehat{\mathrm{g}}(\mathbf{x}) \leq \sqrt{\widehat{\mathrm{g}}(\mathbf{x})} \leq 1$, for all $\mathbf{x} \in \mathcal{X}$, which simply follows using the subadditive of the function $t \mapsto \sqrt{t}$ and the definition of $\widehat{\mathrm{g}}(\mathbf{x})$. By Markov's inequality,

$$\Pr\left(\Delta(\mathbf{X}) \geq \varepsilon(\eta)\right) \leq \eta \quad \text{with} \quad \varepsilon(\eta) = 2\sqrt{2\mathbb{E}_{\mathbf{X}Y}\left[-\log P_{\widehat{Y}|X}(Y|\mathbf{X})\right]/\eta}, \tag{13}$$

for any $\eta > 0$, where $\mathbb{E}_{\mathbf{X}Y}\left[-\log P_{\widehat{Y}|X}(Y|\mathbf{X})\right]$ in (13) is the cross-entropy risk. The latter is expected to be small provided that the model generalizes well. Thus, $\varepsilon(\eta)$ can be expected to be small for a desired confidence $\eta > 0$. Interestingly, (11) turns out to be related to the uncertainty of the classifier via the quadratic Rényi entropy [31]: $-\log_2\left(\widehat{\mathrm{g}}(\mathbf{x})\right) = 2H_2(\widehat{Y}|\mathbf{x}) \leq 2H(\widehat{Y}|\mathbf{x})$, where the latter is the Shannon entropy, i.e., the self-uncertainty of the classifier.

3.4 From the theory to a practical discriminator

Our previous discussion suggests that $\widehat{\text{Pe}}(\mathbf{x})$ in (2) may be a valuable candidate to approximate $\text{Pe}(\mathbf{x})$ in the definition of the optimal discriminator (10). On the other hand, Proposition 3.2 suggests that $1 - \widehat{\text{g}}(\mathbf{x})$ can also be a valuable candidate yielding another discriminator. These discriminators, referred to as DOCTOR, are introduced below.

Definition 2 (DOCTOR)**.** For any $0 < \gamma < \infty$ and $\mathbf{x} \in \mathcal{X}$, define the following discriminators:

$$D_\alpha(\mathbf{x}, \gamma) \triangleq \mathbb{1}\left[1 - \widehat{\text{g}}(\mathbf{x}) > \gamma \cdot \widehat{\text{g}}(\mathbf{x})\right], \qquad D_\beta(\mathbf{x}, \gamma) \triangleq \mathbb{1}\left[\widehat{\text{Pe}}(\mathbf{x}) > \gamma \cdot (1 - \widehat{\text{Pe}}(\mathbf{x}))\right]. \tag{14}$$

Notice that because of Definition 2 and (11), $D_\alpha(\mathbf{x}, \gamma) = \mathbb{1}[1 - \sum_{y \in \mathcal{Y}} \text{softmax}^2(\mathbf{x})_y > \gamma \cdot \sum_{y \in \mathcal{Y}} \text{softmax}^2(\mathbf{x})_y]$; similarly because of Definition 2 and eq. (2), $D_\beta(\mathbf{x}, \gamma) = \mathbb{1}[1 - \max_{y \in \mathcal{Y}} \text{softmax}(\mathbf{x})_y > \gamma \cdot \max_{y \in \mathcal{Y}} \text{softmax}(\mathbf{x})_y]$. The performance of these discriminators will be investigated and compared to state-of-the-art methods in the next section. In the supplementary material (Appendix B), we illustrate how DOCTOR and the optimal discriminator (Definition 1) work on a synthetic data model that is a mixture of two spherical Gaussians with one component per class.

4 Experimental Results

In this section we present a collection of experimental results to investigate the effectiveness of DOCTOR, by applying it to several benchmark datasets. We provide publicly available code[1] to reproduce our results, and we give further details on the environment, the parameter setting and the experimental setup in the Supplementary material (Appendix C). We propose a comparison with state-of-the-art methods using similar information. Though we are not concerned with the OOD detection problem, we are confident it is appropriate to compare DOCTOR to methods which use soft-probabilities or at most the output of the latent code, e.g., ODIN [23], softmax response (SR) [6] and Mahalanobis distance (MHLNB) [22]. Since we are focusing on misclassification detection, it is expected that OOD samples should be also detected as classification errors.

Totally Black Box (TBB) and Partially Black Box (PBB). We address two different scenarios with respect to the available information about the network. In the TBB only the output of the last layer of the network is available, hence gradient-propagation to perform input pre-processing is not allowed. In the PBB we allow method-specific inputs perturbations. When considering DOCTOR in PBB, for each testing sample $\mathbf{x}$, we calculate the pre-processed sample $\widetilde{\mathbf{x}}$ by adding a small perturbation:

$$\widetilde{\mathbf{x}}^\alpha = \mathbf{x} - \epsilon \times \text{sign}\left[-\nabla_{\mathbf{x}} \log\left(\frac{1 - \widehat{\text{g}}(\mathbf{x})}{\widehat{\text{g}}(\mathbf{x})}\right)\right], \text{ and } \widetilde{\mathbf{x}}^\beta = \mathbf{x} - \epsilon \times \text{sign}\left[-\nabla_{\mathbf{x}} \log\left(\frac{\widehat{\text{Pe}}(\mathbf{x})}{1 - \widehat{\text{Pe}}(\mathbf{x})}\right)\right].$$

We will write directly $\widetilde{\mathbf{x}}$ when it is clear from the context which input pre-processing we are referring to. In Supplementary material (Appendix C.2) we further analyze the equations above. When ODIN or MHLNB are used, we pre-process the inputs as in [23] and in [22], respectively.

4.1 Review of related methods

PBB. We compare DOCTOR (using input pre-processing and temperature scaling) with ODIN and MHLNB. ODIN [23] comprises temperature scaling and input pre-processing via perturbation. Temperature scaling is applied to its scoring function, which has $f_i(\widetilde{\mathbf{x}})$ for the logit of the i-th class. Formally, given an input sample $\mathbf{x}$:

$$\text{SODIN}(\widetilde{\mathbf{x}}) = \max_{i=[1:C]} \frac{\exp(f_i(\widetilde{\mathbf{x}})/T)}{\sum_{j=1}^{C} \exp(f_j(\widetilde{\mathbf{x}})/T)}, \ \text{ODIN}(\widetilde{\mathbf{x}}; \delta, T, \epsilon) = \begin{cases} \text{out}, & \text{if SODIN}(\widetilde{\mathbf{x}}) \leq \delta \\ \text{in}, & \text{if SODIN}(\widetilde{\mathbf{x}}) > \delta, \end{cases}$$

where $\widetilde{\mathbf{x}}$ represents a magnitude ϵ perturbation of the original $\mathbf{x}$; T is the temperature scaling parameter; $\delta \in [0, 1]$ is the threshold value; in indicates the acceptance decision while out indicates the rejection decision. Notice, however, γ in DOCTOR and δ in ODIN, respectively, are defined over two different domains: if δ denotes a probability, γ is a ratio between probabilities. Although ODIN

[1] `https://github.com/doctor-public-submission/DOCTOR/`

originally required tuning the hyper-parameter T with out-of-distribution data, it was also shown that a large value for T is generally desirable, suggesting that this gain is achieved at 1000. Anyway, in this framework, we notice an improvement of ODIN in performance for low values of T. Thus we report the best results obtained by ODIN considering the range of hyper-parameters values tested also for DOCTOR (cf. section 4.3). ENERGY [24] comprises the denominator of the softmax activation:

$$\text{ES}(\mathbf{x}) = -T \cdot \log \sum_{j=1}^{C} \exp(f_j(\mathbf{x})/T), \ \text{ENERGY}(\mathbf{x}; \xi, T) = \begin{cases} \text{out}, & \text{if } -\text{ES}(\mathbf{x}) \leq \xi \\ \text{in}, & \text{if } -\text{ES}(\mathbf{x}) > \xi, \end{cases}$$

where $\xi \in \mathbb{R}$ is the threshold value. Unlike all the methods considered in this paper, MHLNB [22] requires the knowledge of the training set $\mathcal{D}_n$ which the pre-trained network was trained on to compute its *empirical class mean* $\widehat{\mu}_c$ for each class c and its *empirical covariance* $\widehat{\Sigma}$:

$$\widehat{\mu}_c = \frac{1}{n_c} \sum_{i:\, y_i = c} f(\widetilde{\mathbf{x}}_i); \ \widehat{\Sigma} = \frac{1}{n} \sum_{c \in \mathcal{Y}} \sum_{\forall i:\, y_i = c} (f(\widetilde{\mathbf{x}}_i) - \widehat{\mu}_c)(f(\widetilde{\mathbf{x}}_i) - \widehat{\mu}_c)^\top,$$

where n_c denotes the number of training samples with label c and $f(\widetilde{\mathbf{x}})$ the logits vector. As MHLNB directly uses the vector of logits, it does not comprise temperature scaling. Given an input sample $\mathbf{x}$:

$$\text{M}(\widetilde{\mathbf{x}}) = \max_{c \in \mathcal{Y}} \ -(f(\widetilde{\mathbf{x}}) - \widehat{\mu}_c)^\top \widehat{\Sigma}^{-1} (f(\widetilde{\mathbf{x}}) - \widehat{\mu}_c), \ \text{MHLNB}(\widetilde{\mathbf{x}}; \zeta, \epsilon) = \begin{cases} \text{out}, & \text{if M}(\widetilde{\mathbf{x}}) > \zeta \\ \text{in}, & \text{if M}(\widetilde{\mathbf{x}}) \leq \zeta, \end{cases}$$

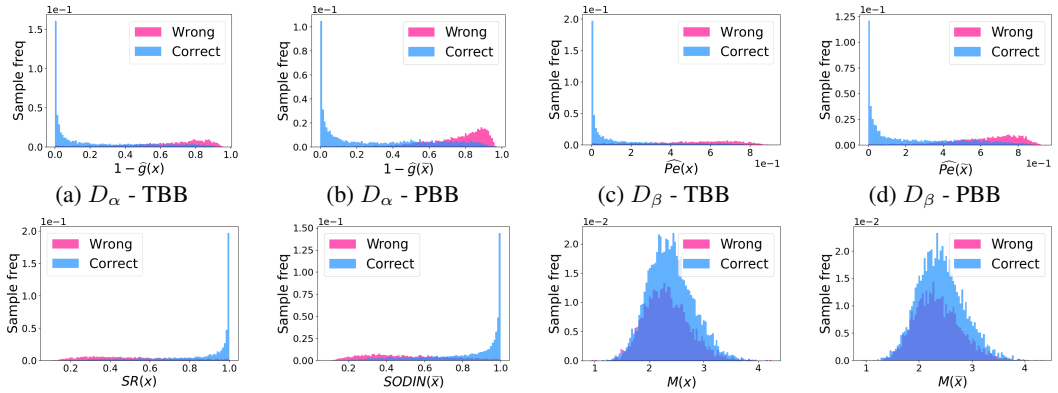
as mentioned above, $\widetilde{\mathbf{x}}$ represents a magnitude ϵ perturbation of the original $\mathbf{x}$; $\zeta \in \mathbb{R}_+$ is the threshold value; in indicates the acceptance decision while out indicates the rejection decision.

TBB. We compare DOCTOR (without input pre-processing and temperature scaling) with MHLNB (without input pre-processing and with the softmax output layer in place of the logits) and SR. Although both DOCTOR and SR have access to the softmax output of the predictor, a fundamental difference is that, while the former utilizes the softmax output in its entirety, the latter only uses the maximum value, therefore discarding potentially useful information. As it will be shown, this leads to better results for DOCTOR on several datasets (see table 1). We emphasize that, by setting $T = 1$ and $\epsilon = 0$, ODIN reduces to softmax response [6] since $\text{SR}(\mathbf{x}) \equiv \text{SODIN}(\mathbf{x})$.

4.2 Detection of misclassification errors, experimental setup and evaluation metrics

Before digging into the detailed discussion of our numerical results, we present an empirical analysis of the behavior of DOCTOR, ODIN, SR and MHLNB when faced with the task of choosing whether to accept or reject the prediction of a given classifier for a certain sample. In Figure 1, we propose a graphical interpretation of the discrimination performance, considering the labeled samples in the dataset TinyImageNet and the ResNet network as the classifier. We separate correctly and incorrectly classified samples according to their true labels in blue and in red, respectively. We remind that the label information is *not* necessary for the discriminators to define acceptance and rejection regions. Then, for each sample we compute the corresponding discriminators' output. These values are binned and reported on the horizontal axis of Figure 1a and Figure 1b for D_α, Figure 1c and Figure 1d for D_β, Figure 1e for SR, Figure 1f for ODIN, Figure 1g and Figure 1h for MHLNB. In each each plot, and according to the corresponding discriminator, the bins' heights represent the frequency of the samples whose value falls within that bin. The intuition is that, if moving along the horizontal axis it is possible to pick a threshold value such that, w.r.t. this value, blue bars are on one side of the plot and red bars on the other, this threshold would correspond to the optimal discriminator, i.e. the discriminator that chooses the optimal acceptance and rejection regions.

In Figure 1g through Figure 1h, we observe that, for MHLNB, no matter how well we choose the threshold value, it is hard to fully separate red and blue bars both in TBB and PBB, i.e. the discriminator fails at defining acceptance and rejection regions so that all the hits can be assigned to the first one and all the mis-classification to the second one. The samples distribution for SR and ODIN in Figure 1e and Figure 1f, respectively, does not look significantly different from the one related to D_α and D_β in TBB (Equation (14)). However, the discrimination between samples becomes evident in PBB. This is shown in Figure 1d for D_β (eq. (14)) and even more in Figure 1b for D_α (eq. (14)) where, quite clearly, rightly classified samples are clustered on the left-end side of the plot and incorrectly classified samples tend to cluster on the right-end side. This intuition is supported by the results in Table 1.



(a) D_α - TBB (b) D_α - PBB (c) D_β - TBB (d) D_β - PBB

(e) SR - TBB (f) ODIN - PBB (g) MHLNB - TBB (h) MHLNB - PBB

Figure 1: DOCTOR, ODIN, SR and MHLNB to split data samples in TinyImageNet both under TBB and PBB: (a) - (b) show the results for expressions (2); (c) - (d) show the results for (11); (e) shows the results for SR; (f) shows the results for ODIN; (g) - (h) show the results for MHLNB. Histograms for wrongly classified samples (red) and correctly classified samples (blue).

Datasets and pre-trained networks. We run experiments on both image and textual datasets. We use CIFAR10 and CIFAR100 [18], TinyImageNet [16] and SVHN [27] as image datasets; IMDb [25], AmazonFashion and AmazonSoftware [28] as textual datasets. Note that, for all the aforementioned datasets, we consider only the test set since we rely on pre-trained models. Along the same lines of [23], we use the pre-trained DenseNet models [15] for CIFAR10, CIFAR100 and SVHN. In addition, we use a pre-trained ResNet model [10] for TinyImageNet, and BERT [3, 34] for the Amazon datasets and IMDb. The accuracy achieved by the aforementioned networks on the test sets is showed in Table 1. According to the invariant properties of the discriminator (see Def. 2) with respect to the soft-probability of the underlying model, permutations of the posterior probabilities vector, due different initialization of the models before the training, do not change the output of Eq. (10), as it is a sum of squared values of the softmax probabilities. This variety of models/datasets characterizes the performance of the proposed method in scenarios with different accuracy levels.

Evaluation metrics. We will evaluate the performance according to Proposition (3.1) via the empirical estimates of Type-I and Type-II errors in expressions (5). Throughout this section, when the model's decision for a sample is correct (hit) but is rejected by the discriminator, we refer to such event as *false rejection*; when the model's decision for a sample is not correct (miss) and is rejected by the discriminator, we refer to such event as *true rejection*. Similarly, we refer to a *false acceptance* when a miss is not rejected and to a *true acceptance* when a hit is not rejected. More specifically, let $\mathcal{T}_m = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\} \sim p_{XY}$ be the *testing set*, where $\mathbf{x}_i \in \mathcal{X}$ is the input sample, $y_i \in \{1, \ldots, C\}$ is the true class of $\mathbf{x}_i$, and m denotes the size of the testing set. With $j \in \{\alpha, \beta\}$:

$$\mathcal{FR}_j(\gamma) = \{(\mathbf{x}, y) \in \mathcal{T}_m : y = f_{\mathcal{D}_n}(\mathbf{x}),\ D_j(\mathbf{x}, \gamma) = 1\}, \tag{15}$$

$$\mathcal{TR}_j(\gamma) = \{(\mathbf{x}, y) \in \mathcal{T}_m : y \neq f_{\mathcal{D}_n}(\mathbf{x}),\ D_j(\mathbf{x}, \gamma) = 1\}, \tag{16}$$

$$\mathcal{FA}_j(\gamma) = \{(\mathbf{x}, y) \in \mathcal{T}_m : y \neq f_{\mathcal{D}_n}(\mathbf{x}),\ D_j(\mathbf{x}, \gamma) = 0\}, \tag{17}$$

$$\mathcal{TA}_j(\gamma) = \{(\mathbf{x}, y) \in \mathcal{T}_m : y = f_{\mathcal{D}_n}(\mathbf{x}),\ D_j(\mathbf{x}, \gamma) = 0\}\,. \tag{18}$$

We measure the performance of the test in terms of:

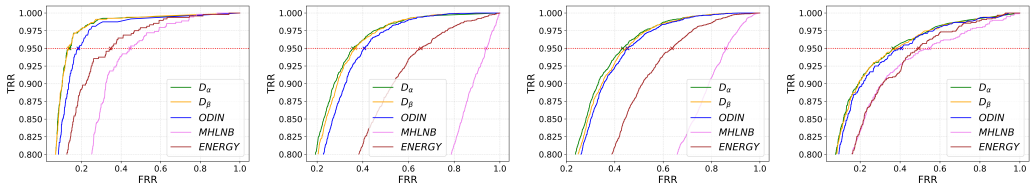
- **FRR** versus **TRR**. The false rejection rate (FRR) represents the probability that a hit is rejected, while the true rejection rate (TRR) is the probability that a miss is rejected.
- **AUROC**. The area under the *Receiver Operating Characteristic curve* (ROC) [2] depicts the relationship between TRR and FRR. The perfect detector corresponds to a score of 100%.
- **FRR at 95 % TRR.** This is the probability that a hit is rejected when the TRR is at 95 %.

4.3 Experimental results: comparison between different discriminators

DOCTOR: comparison between D_α and D_β. We compare the discriminators D_α and D_β introduced in (14) to show how the AUROCs for CIFAR10, CIFAR100, TinyImageNet and SVHN change when

varying the parameters T and ϵ. It is observed that D_α is less sensitive to the selection of T: for all the datasets, D_α outperforms D_β achieving the best AUROCs by setting $T = 1$. Contrary to D_α, D_β is more sensitive to the value selected for T in the sense that small changes may result in very different values for the measured AUROCs (cf. Appendix C.4.1). In contrast,the best results are obtained for the same epsilon values of D_α and D_β across all the datasets.

Comparison in TBB. We compare DOCTOR with MHLNB (without input pre-processing and with the softmax output in place of the logits) and SR. It is worth to emphasize that D_α does not coincide (in general) with SR since the former consists in the sum of squared values of all probabilities involved in the softmax. To complete the comparison, we include the results for both methods in Table 1.



(a) CIFAR10 - PBB (b) CIFAR100 - PBB (c) TinyImageNet - PBB (d) SVHN - PBB

Figure 2: ROC curves. Comparison between D_α ($T_\alpha = 1$ and $\epsilon_\alpha = 0.00035$), D_β ($T_\beta = 1.5$ and $\epsilon_\alpha = 0.00035$), ODIN ($T_{\text{ODIN}} = 1.3$ and $\epsilon_{\text{ODIN}} = 0$), MHLNB ($T_{\text{MHLNB}} = 1$ and $\epsilon_{\text{MHLNB}} = 0.0002$) and ENERGY ($T_{\text{ENERGY}} = 1$ and $\epsilon_{\text{ENERGY}} = 0$). Red dashed lines mark the 95% threshold of TRR.

Comparison in PBB. We compare DOCTOR with ODIN, MHLNB and ENERGY. We keep the same parameter setting for all the methods. In the case of DOCTOR and ODIN where temperature scaling is allowed, we test, for each dataset, 24 different values of ϵ for each of the 11 different values of T, see (Appendix C.4.2) for the set of ranges. In the case of MHLNB, which directly uses the logits, we keep $T = 1$ and we vary ϵ for each dataset. In the case of ENERGY, where no perturbation is allowed, we keep $\epsilon = 0$ and we maintain $T = 1$ (as in [24]). According to our framework, no validation samples are available; consequently, in order to be consistent across the datasets, we only report the experimental settings and values for which, on average, we obtain favorable results for all the considered domains (cf. Figure 2). In order to be fair, we update ODIN's parameters from those in [23] to new values which are more suitable to the task at hand (cf. plots in Appendix C.4.2).

DOCTOR's performance compared to ODIN's, MHLNB's and ENERGY's, are collected in Table 1 and in Figure 2. The results in the table show that noise further improves the performance of DOCTOR (cf. PBB) up to 1% over our previous experiments without noise (cf. TBB) in terms of AUROC. The improvement is even more significant in terms of FRR at 95% TRR: *a 4% decrease is obtained in terms of predictions incorrectly rejected for* DOCTOR *when passing from TBB to PBB*. Note that only the softmax output is available when we consider the pre-trained models for AmazonFashion, AmazonSoftware and IMDb datasets; therefore, we cannot access any internal layer and test DOCTOR for values of T which differ from the default value $T = 1$. Consequently, temperature scaling and input pre-processing cannot be applied in these cases and thus these datasets cannot be tested in PBB. Moreover, even in TBB, these datasets cannot be tested through MHLNB and ENERGY since the dataset on which the network was trained is not available. We provide simulations on how the range of interval for the different thresholds can affect the results in Appendix C.3.

Misclassification detection in presence of OOD samples. We evaluate DOCTOR's performance in misclassifcation detection considering a mixture of both in (DATASET-IN) and out-of-distribution (OOD) samples (DATASET-OUT), i.e. input samples for which the decision should not be trusted. The results are compared with ODIN. We test the two methods when one sample to reject out of five (♣), three (♢) or two (♠) is OOD. The details of the simulations, the considered dataset, and the complete experimental results are relegated Appendix C.4.3. In Table 2 we report an extract of the results for the PBB scenario in terms of *mean / standard deviation*: DOCTOR achieves, and most of the time outperforms ODIN's performance. We emphasize that, even though DOCTOR is not tuned for the OOD detection problem, it represents the best choice for deciding whether to accept or reject the prediction of the classifier also on mixed data scenarios where the percentage of OOD samples, as long as it is not dominant, can sensitively vary.

Table 1: For all methods, in TBB, we set $T = 1$ and $\epsilon = 0$; in PBB we set : $\epsilon_\alpha = \epsilon_\beta = 0.00035$, $T_\alpha = 1$, $T_\beta = 1.5$, $\epsilon_{\text{ODIN}} = 0$ and $T_{\text{ODIN}} = 1.3$, $\epsilon_{\text{MHLNB}} = 0.0002$ and $T_{\text{MHLNB}} = 1$, $\epsilon_{\text{ENERGY}} = 0$ and $T_{\text{ENERGY}} = 1$. In TBB, ODIN and SR coincide ($T = 1$ and $\epsilon = 0$).

DATASET	METHOD	AUROC %		FRR % (95 % TRR)	
		TBB	PBB	TBB	PBB
CIFAR10 Acc. 95%	D_α	**94**	**95.2**	**17.9**	13.9
	D_β	68.5	94.8	18.6	**13.4**
	ODIN	93.8	94.2	18.2	18.4
	SR	93.8	-	18.2	-
	MHLNB	92.2	84.4	30.8	44.6
	ENERGY	-	91.1	-	34.7
CIFAR100 Acc. 78%	D_α	**87**	**88.2**	40.6	**35.7**
	D_β	84.2	87.4	40.6	36.7
	ODIN	86.9	87.1	40.5	40.7
	SR	86.9	-	**40.5**	-
	MHLNB	82.6	50	66.7	94
	ENERGY	-	78.7	-	65.4
TINY IMAGENET Acc. 63%	D_α	**84.9**	**86.1**	**45.8**	**43.3**
	D_β	**84.9**	85.3	**45.8**	45.1
	ODIN	84.9	84.9	45.8	45.3
	SR	**84.9**	-	**45.8**	-
	MHLNB	78.4	59	82.3	86
	ENERGY	-	78.2	-	63.7

DATASET	METHOD	AUROC %		FRR % (95 % TRR)	
		TBB	PBB	TBB	PBB
SVHN Acc. 96%	D_α	**92.3**	**93**	**38.6**	**36.6**
	D_β	92.2	92.8	39.7	38.4
	ODIN	92.3	92.3	38.6	40.7
	SR	**92.3**	-	**38.6**	-
	MHLNB	87.3	88	85.8	54.7
	ENERGY	-	88.9	-	49.4
AMAZON FASHION Acc. 85%	D_α	**89.7**	-	27.1	-
	D_β	**89.7**	-	**26.3**	-
	SR	87.4	-	50.1	-
AMAZON SOFTWARE Acc. 73%	D_α	**68.8**	-	**73.2**	-
	D_β	**68.8**	-	**73.2**	-
	SR	67.3	-	86.6	-
IMDB Acc. 90%	D_α	**84.4**	-	**54.2**	-
	D_β	**84.4**	-	54.4	-
	SR	83.7	-	61.7	-

Table 2: Same parameter setting as in table 1 (PBB) for D_α, D_β, ODIN, ENERGY; as in [23] for ODIN_{OOD} and as in [22] for MHLNB_{WB}. Results presented in terms of *mean / standard deviation*.

DATASET-IN	DATASET-OUT	AUROC %						FRR % (95 % TRR)					
		D_α	D_β	ODIN	ODIN_{OOD}	ENERGY	MHLNB_{WB}	D_α	D_β	ODIN	ODIN_{OOD}	ENERGY	MHLNB_{WB}
CIFAR10 ♣	iSUN	**95.4** / 0.1	95.1 / 0.1	94.6 / 0.1	89.6 / 0	92.4 / 0.1	54.5 / 0.1	14 / 0.5	**13.5** / 0.4	17.2 / 0.3	38.9 / 0	32.2 / 0.1	92 / 0.1
	TINY (RES)	**95.2** / 0.1	94.9 / 0	94.6 / 0.1	89.6 / 0	92.3 / 0.1	56.2 / 0	**14** / 0.4	**14** / 0.5	17.8 / 0.4	38.9 / 0	32.2 / 0.1	90.3 / 0.2
CIFAR10 ♢	iSUN	**95.5** / 0.1	95.3 / 0.1	94.9 / 0.1	91.5 / 0	92.9 / 0	54.5 / 0.1	14.4 / 0.6	**13.4** / 0.2	16.8 / 0.5	34/ 0.1	27 / 1	92 / 0.2
	TINY (RES)	**95.4** / 0.1	95 / 0.1	94.8 / 0.1	91.4 / 0	92.8 / 0	56.2 / 0.1	15 / 0.1	**14.8** / 0.7	17 / 0.5	34.5 / 0.9	28.8 / 1.9	90 / 0.3
CIFAR10 ♠	iSUN	**95.6** / 0.1	**95.6** / 0	95.4 / 0	93.5 / 0	93.6 / 0.1	54.6 / 0	15.1 / 0.1	**13.6** / 0.5	16.1 / 0.2	30.6 / 0.4	25.1 / 0.2	92 / 0.2
	TINY (RES)	**95.5** / 0.1	95.2 / 0.1	95.1 / 0.1	93.2	93.5 / 0	56.2 / 0.2	**14.7** / 0.3	14.8 / 0.5	17.1 / 0.4	31/ 0	25.6 / 0.3	90.2 / 0.1

5 Summary and Concluding Remarks

We introduced a simple and effective method to detect misclassification errors, i.e., whether a prediction of a classifier should or should not be trusted. We provided theoretical results on the optimal statistical model for misclassification detection and we presented our empirical discriminator DOCTOR. Experiments on real (textual and visual) datasets–including OOD samples and comparison to state-of-the-art methods– demonstrate the effectiveness of our proposed methods. Whilst methods for ODD frameworks do not necessarily perform well in predicting misclassification errors, our result advances the state-of-the-art, and the main takeaway is that DOCTOR can be applied to both partially black-box (PBB) setups and totally black-box (TBB) ones. In the latter, information about the model's architecture may be undisclosed for security reason when dealing with sensitive data). DOCTOR uses all the information in the softmax output, which results in equal or better performance with respect to the other methods: the results in PBB, where we observe a reduction up to 4% in terms of predictions incorrectly rejected with respect to the ones in TBB are particularly promising. Moreover, DOCTOR does not require training data and, thanks to its flexibility, it can be easily deployed in real-world scenarios. Currently, DOCTOR does not exploit information across the layers yet. Only the soft-predictions are used. Besides, the most important obstacle is the calibration of the threshold (γ) between the desired fault rejection and acceptance rates, which would require additional validation samples. However, quite often, the cost of collecting data for this operation can be prohibitive, making it difficult or too expensive to perform such calibration. As future work, we shall combine DOCTOR with other related lines of research such as: the extension to white-box incorporating additional information across the different latent codes of the model. Moreover, we shall investigate the possibility of combining the two proposed discriminators.

References

[1] J. Chen, Y. Li, X. Wu, Y. Liang, and S. Jha. Robust out-of-distribution detection in neural networks. *arXiv preprint arXiv:2003.09711*, 2020.

[2] J. Davis and M. Goadrich. The relationship between precision-recall and ROC curves. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, volume 148, pages 233–240, 2006.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[4] T. DeVries and G. W. Taylor. Learning confidence for out-of-distribution detection in neural networks. *CoRR*, abs/1802.04865, 2018.

[5] A. Gangrade, A. Kag, and V. Saligrama. Selective classification via one-sided prediction. In A. Banerjee and K. Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 2179–2187. PMLR, 2021.

[6] Y. Geifman and R. El-Yaniv. Selective classification for deep neural networks. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4878–4887, 2017.

[7] Y. Geifman and R. El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97, pages 2151–2159, 2019.

[8] Y. Geifman, G. Uziel, and R. El-Yaniv. Reduced uncertainty estimation for deep neural classifiers. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[9] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70, pages 1321–1330, 2017.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[11] M. Hein, M. Andriushchenko, and J. Bitterwolf. Why networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 41–50. Computer Vision Foundation / IEEE, 2019.

[12] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[13] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[14] Y. Hsu, Y. Shen, H. Jin, and Z. Kira. Generalized ODIN: detecting out-of-distribution image without learning from out-of-distribution data. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10948–10957, 2020.

[15] G. Huang, Z. Liu, and K. Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[16] Q. Z. Jiayu Wu and G. Xu. Tiny imagenet challenge. Technical report, 2017.

[17] A. Kristiadi, M. Hein, and P. Hennig. Being bayesian, even just a bit, fixes overconfidence in relu networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5436–5446. PMLR, 2020.

[18] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[19] V. Kuleshov and S. Ermon. Reliable confidence estimation via online learning. *CoRR*, abs/1607.03594, 2016.

[20] V. Kuleshov and P. Liang. Calibrated structured prediction. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3474–3482, 2015.

[21] K. Lee, H. Lee, K. Lee, and J. Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[22] K. Lee, K. Lee, H. Lee, and J. Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7167–7177, 2018.

[23] S. Liang, Y. Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[24] W. Liu, X. Wang, J. D. Owens, and Y. Li. Energy-based out-of-distribution detection. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[25] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, June 2011.

[26] A. Meinke and M. Hein. Neural networks that provably know when they don't know. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

[27] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Ng. Reading digits in natural images with unsupervised feature learning. *NIPS*, 01 2011.

[28] J. Ni, J. Li, and J. J. McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 188–197, 2019.

[29] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Adv. Large Margin Classif.*, 10, 06 2000.

[30] A. B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Publishing Company, Incorporated, 1st edition, 2008.

[31] T. van Erven and P. Harremos. Rényi Divergence and Kullback-Leibler Divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

[32] V. Vapnik and R. Izmailov. Complete statistical theory of learning: learning using statistical invariants. In *Proceedings of the Ninth Symposium on Conformal and Probabilistic Prediction and Applications*, volume 128, pages 4–40, 2020.

[33] A. Vyas, N. Jammalamadaka, X. Zhu, D. Das, B. Kaul, and T. L. Willke. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VIII*, volume 11212, pages 560–574, 2018.

[34] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, oct 2020.

[35] C. Xing, S. Arik, Z. Zhang, and T. Pfister. Distance-based learning from errors for confidence calibration. In *International Conference on Learning Representations*, 2020.

[36] Z. Zhang, A. V. Dalca, and M. R. Sabuncu. Convolutional neural networks using structured dropout. *CoRR*, abs/1906.09551, 2019.

Supplementary Material

A Proofs

The following section shows the proofs for Proposition (3.1), Proposition (3.2) and Inequalities (3).

A.1 Proof of Proposition 3.1

We recall the definition of the total variation distance when applied to distributions P, Q on a set $\mathcal{X} \subseteq \mathbb{R}^d$ and the Scheffé's identity, Lemma 2.1 in [30]:

$$\|P - Q\|_{\text{TV}} \triangleq \sup_{\mathcal{A} \in \mathcal{B}^d} |P(\mathcal{A}) - Q(\mathcal{A})| = \frac{1}{2} \int |p_X(\mathbf{x}) - q_X(\mathbf{x})| d\mu(\mathbf{x}) \tag{19}$$

with respect to a base measure μ, where $\mathcal{B}^d$ denotes the class of all Borel sets on $\mathbb{R}^d$.

Proof. First of all, we prove the equality for $\gamma = 1$. Let us denote with $\mathcal{A}^\star \equiv \mathcal{A}(1)$ and $\mathcal{A}^{\star c} \equiv \mathcal{A}^c(1)$ the optimal decision regions from (9). Let $\epsilon_0(\mathcal{A}^\star)$ and $\epsilon_1(\mathcal{A}^{\star c})$ be the Type-I and Type-II errors, respectively. Then,

$$\begin{aligned}
\epsilon_0(\mathcal{A}^\star) + \epsilon_1(\mathcal{A}^{\star c}) &= \int_{\mathcal{A}^\star} p_{X|E}(\mathbf{x}|0) d\mathbf{x} + \int_{\mathcal{A}^{\star c}} p_{X|E}(\mathbf{x}|1) d\mathbf{x} \\
&= \int_{\mathcal{A}^\star} \min\Big\{ p_{X|E}(\mathbf{x}|0)\,,\, p_{X|E}(\mathbf{x}|1) \Big\} d\mathbf{x} \\
&+ \int_{\mathcal{A}^{\star c}} \min\Big\{ p_{X|E}(\mathbf{x}|0)\,,\, p_{X|E}(\mathbf{x}|1) \Big\} d\mathbf{x} \\
&= \int_{\mathcal{X}} \min\Big\{ p_{X|E}(\mathbf{x}|0)\,,\, p_{X|E}(\mathbf{x}|1) \Big\} d\mathbf{x} \\
&= 1 - \big\| p_{X|E=1} - p_{X|E=0} \big\|_{\text{TV}}\,,
\end{aligned} \tag{20}$$

where the last identity follows by applying Scheffé's identity (19). From the last identity in (20) and any decision region $\mathcal{A} \subseteq \mathcal{X}$, we have

$$\begin{aligned}
1 - \big\| p_{X|E=1} - p_{X|E=0} \big\|_{\text{TV}} &= \int_{\mathcal{X}} \min\Big\{ p_{X|E}(\mathbf{x}|0)\,,\, p_{X|E}(\mathbf{x}|1) \Big\} d\mathbf{x} \\
&= \int_{\mathcal{A}} \min\Big\{ p_{X|E}(\mathbf{x}|0)\,,\, p_{X|E}(\mathbf{x}|1) \Big\} d\mathbf{x} \\
&+ \int_{\mathcal{A}^c} \min\Big\{ p_{X|E}(\mathbf{x}|0)\,,\, p_{X|E}(\mathbf{x}|1) \Big\} d\mathbf{x} \\
&\leq \int_{\mathcal{A}} p_{X|E}(\mathbf{x}|0) d\mathbf{x} + \int_{\mathcal{A}^c} p_{X|E}(\mathbf{x}|1) d\mathbf{x} \\
&= \epsilon_0(\mathcal{A}) + \epsilon_1(\mathcal{A}^c).
\end{aligned} \tag{21}$$

It remains to show the last statement related to the Bayesian error of the test. Assume that $P_E(1) = P_E(0) = 1/2$. By using the last identity in (20), we have

$$\begin{aligned}
\frac{1}{2}\Big[1 - \big\| p_{X|E=1} - p_{X|E=0} \big\|_{\text{TV}} \Big] &= \frac{1}{2} \int_{\mathcal{X}} \min\Big\{ p_{X|E}(\mathbf{x}|0), p_{X|E}(\mathbf{x}|1) \Big\} d\mathbf{x} \\
&= \int_{\mathcal{X}} \min\Big\{ p_{XE}(\mathbf{x}, E=0), p_{XE}(\mathbf{x}, E=1) \Big\} d\mathbf{x} \\
&= \mathbb{E}_X\left[\min\Big\{ P_{E|X}(0|\mathbf{X}), P_{E|X}(1|\mathbf{X}) \Big\} \right] \\
&= \frac{1}{2}\big[\epsilon_0(\mathcal{A}^\star) + \epsilon_1(\mathcal{A}^{\star c}) \big] \\
&\equiv \inf_D \Pr\left\{ D(\mathbf{X}) \neq E \right\},
\end{aligned} \tag{22}$$

where the last identity follow by the definition of the decision regions in (9). □

A.2 Proof of Proposition 3.2

Proof. We begin by showing that

$$\begin{aligned}
|\widehat{\mathrm{Pe}}(\mathbf{x}) - \mathrm{Pe}(\mathbf{x})| &= \Big|\mathbb{E}\big[\mathbb{1}[\widehat{Y} \neq f_{\mathcal{D}_n}(\mathbf{x})]\big|\mathbf{x}\big] - \mathbb{E}\big[\mathbb{1}[Y \neq f_{\mathcal{D}_n}(\mathbf{x})]\big|\mathbf{x}\big]\Big| \\
&= \left|\sum_{\{y\in\mathcal{Y}\,|\,y\neq f_{\mathcal{D}_n}(\mathbf{x})\}} \big[P_{\widehat{Y}|X}(y|\mathbf{x}) - P_{Y|X}(y|\mathbf{x})\big]\right| \\
&\leq \sum_{\{y\in\mathcal{Y}\,|\,y\neq f_{\mathcal{D}_n}(\mathbf{x})\}} \Big|P_{\widehat{Y}|X}(y|\mathbf{x}) - P_{Y|X}(y|\mathbf{x})\Big| \\
&\leq \sum_{y\in\mathcal{Y}} \Big|P_{\widehat{Y}|X}(y|\mathbf{x}) - P_{Y|X}(y|\mathbf{x})\Big| \\
&\leq 2\,\Big\|P_{\widehat{Y}|X}(\cdot|\mathbf{x}) - P_{Y|X}(\cdot|\mathbf{x})\Big\|_{\mathrm{TV}} \\
&\leq 2\sqrt{2\mathrm{KL}\left(P_{Y|\mathbf{x}}\|P_{\widehat{Y}|\mathbf{x}}\right)}, \qquad (23)
\end{aligned}$$

where $\|\cdot\|_{\mathrm{TV}}$ denotes the *Total Variation distance*, $\mathrm{KL}(\cdot\|\cdot)$ is the *Kullback–Leibler divergence* and the last step is due to *Pinsker's inequality*. On the other hand,

$$\begin{aligned}
1 - \widehat{\mathrm{g}}(\mathbf{x}) &= 1 - \sum_{y\in\mathcal{Y}} P^2_{\widehat{Y}|X}(y|\mathbf{x}) \\
&= 1 - \mathbb{E}_{\widehat{Y}|X}\left[P_{\widehat{Y}|X}(\widehat{Y}|\mathbf{x})|\mathbf{x}\right] \\
&\geq 1 - \mathbb{E}_{\widehat{Y}|X}\left[\max_{y\in\mathcal{Y}} P_{\widehat{Y}|X}(y|\mathbf{x})\big|\mathbf{x}\right] \\
&= 1 - \max_{y\in\mathcal{Y}} P_{\widehat{Y}|X}(y|\mathbf{x}) \\
&\equiv \widehat{\mathrm{Pe}}(\mathbf{x}). \qquad (24)
\end{aligned}$$

Similarly,

$$\begin{aligned}
\widehat{\mathrm{g}}(\mathbf{x}) &= \sum_{y\in\mathcal{Y}} P^2_{\widehat{Y}|X}(y|\mathbf{x}) \\
&= P^2_{\widehat{Y}|X}(y^\star|\mathbf{x}) + \sum_{y\neq y^\star} P^2_{\widehat{Y}|X}(y|\mathbf{x}) \\
&\geq \max_{y\in\mathcal{Y}} P^2_{\widehat{Y}|X}(y|\mathbf{x}) \\
&\equiv \Big(1 - \widehat{\mathrm{Pe}}(\mathbf{x})\Big)^2, \qquad (25)
\end{aligned}$$

where $y^\star = \arg\max_{y\in\mathcal{Y}} P_{\widehat{Y}|X}(y|\mathbf{x})$. By replacing expressions (24) and (25) in (23) we obtained the desired inequalities, which concludes the proof. □

A.3 Proof of Inequalities in (3)

Proof. The event can be decomposed as follows:

$$\big\{\widehat{E}(\mathbf{x}) \neq E(\mathbf{x})|\mathbf{x}\big\} \equiv \big\{Y \neq \widehat{Y}\big\} \cap \Big\{\big\{\widehat{Y} = f_{\mathcal{D}_n}(\mathbf{x})\big\} \text{ or } \big\{Y = f_{\mathcal{D}_n}(\mathbf{x})\big\}|\mathbf{x}\Big\} \qquad (26)$$

for all $\mathbf{x} \in \mathcal{X}$. Thus,

$$\big\{\widehat{E}(\mathbf{x}) \neq E(\mathbf{x})|\mathbf{x}\big\} \subseteq \big\{Y \neq \widehat{Y}|\mathbf{x}\big\}, \qquad (27)$$

$$\big\{Y \neq \widehat{Y}\big\} \cap \big\{Y \neq f_{\mathcal{D}_n}(\mathbf{x})\big|\mathbf{x}\big\} \subseteq \big\{\widehat{E}(\mathbf{x}) \neq E(\mathbf{x})|\mathbf{x}\big\}, \qquad (28)$$

$$\big\{Y \neq \widehat{Y}\big\} \cap \big\{\widehat{Y} \neq f_{\mathcal{D}_n}(\mathbf{x})|\mathbf{x}\big\} \subseteq \big\{\widehat{E}(\mathbf{x}) \neq E(\mathbf{x})|\mathbf{x}\big\}, \qquad (29)$$

which imply

$$\Pr\left(\{\widehat{E}(\mathbf{x}) \neq E(\mathbf{x})|\mathbf{x}\}\right) \leq \Pr\left(\{\widehat{Y} \neq Y\}|\mathbf{x}\right), \tag{30}$$

$$\mathrm{Pe}(\mathbf{x}) - \Pr\left(\{\widehat{Y} = Y\}|\mathbf{x}\right) \leq \Pr\left(\{\widehat{E}(\mathbf{x}) \neq E(\mathbf{x})\}\big|\mathbf{x}\right), \tag{31}$$

$$\widehat{\mathrm{Pe}}(\mathbf{x}) - \Pr\left(\{\widehat{Y} = Y\}|\mathbf{x}\right) \leq \Pr\left(\{\widehat{E}(\mathbf{x}) \neq E(\mathbf{x})\}\big|\mathbf{x}\right), \tag{32}$$

for all $\mathbf{x} \in \mathcal{X}$, where the last inequalities follows by noticing that $\Pr(\mathcal{A} \cap \mathcal{B}) \geq \Pr(\mathcal{A}) - \Pr(\mathcal{B}^c)$ for arbitrary measurable sets $\mathcal{A}, \mathcal{B} \subset \mathcal{X}$. This concludes the proof of these inequalities. □

B Logistic Regression and Gaussian Model

Throughout this section we test DOCTOR in a controlled setting were all the involved distributions are known. We refer to that setting as *logistic regression and Gaussian model* since we collect data points from Gaussians distributions and we test on the logistic regression setup.

B.1 Theoretical analysis

Let $\mathcal{X} = \mathbb{R}^d$ be the feature space and $\mathcal{Y} = \{-1, 1\}$ be the label space. We focus on a binary classification task in which $\mathbf{X} \sim \mathcal{N}(y\boldsymbol{\mu}, \sigma^2 I)$ and $Y \sim \mathcal{U}(\mathcal{Y})$, where $\boldsymbol{\mu} \in \mathbb{R}^n$ is the mean vector, $\sigma^2 > 0$ is the variance and I is the identity matrix and $\mathcal{U}(\mathcal{Y})$ denotes the uniform distribution over $\mathcal{Y}$. For a fixed $\boldsymbol{\theta} \in \mathbb{R}^d$, consider $f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Y}$ s.t. $f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathrm{sign}(\mathrm{sigmoid}(\mathbf{x}^T\boldsymbol{\theta}) - 1/2)$. For a given $\mathbf{x} \in \mathcal{X}$, we adapt to the current setting the definition of $E(\mathbf{x})$ in section 2 as follows:

$$\mathbb{1}\left[Y \neq f_{\boldsymbol{\theta}}(\mathbf{x})\right] = \mathbb{1}\left[Y \cdot \mathrm{sign}\left(\mathrm{sigmoid}\left(\mathbf{x}^T\boldsymbol{\theta}\right) - \frac{1}{2}\right) < 0\right]. \tag{33}$$

Let us denote by $\mathbb{1}\left[y \neq f_{\boldsymbol{\theta}}(\mathbf{x})\right]$ the realization of the random variable $E(\mathbf{x})$. We can compute the probability of classification error $\mathrm{Pe}(\mathbf{x})$ in (1) w.r.t. the true class posterior probabilities:

$$\begin{aligned}
\mathrm{Pe}(\mathbf{x}) = \mathbb{E}\Big[\mathbb{1}\left[Y \neq f_{\boldsymbol{\theta}}(\mathbf{x})\right]|\mathbf{x}\Big] &= \sum_{y\in\mathcal{Y}} \mathbb{1}\left[y \neq f_{\boldsymbol{\theta}}(\mathbf{x})\right] \cdot \frac{p_{\mathbf{X}|Y}(\mathbf{x}|y)P_Y(y)}{p_{\mathbf{X}}(\mathbf{x})} \\
&= \sum_{y\in\mathcal{Y}} \mathbb{1}\left[y \neq f_{\boldsymbol{\theta}}(\mathbf{x})\right] \cdot \frac{\frac{1}{2}\mathcal{N}(\mathbf{x}; y\mu, \sigma^2 I)}{\frac{1}{2}\sum_{y'\in\mathcal{Y}} \mathcal{N}(\mathbf{x}; y'\mu, \sigma^2 I)} \\
&= \frac{\sum_{y\in\mathcal{Y}} \mathbb{1}\left[y \neq f_{\boldsymbol{\theta}}(\mathbf{x})\right] \cdot \mathcal{N}(\mathbf{x}; y\mu, \sigma^2 I)}{\sum_{y\in\mathcal{Y}} \mathcal{N}(\mathbf{x}; y\mu, \sigma^2 I)}.
\end{aligned} \tag{34}$$

Following (10), the decision region corresponding to the most powerful discriminator for the logistic regression and the Gaussian model are given by

$$\mathcal{A}(\gamma) = \left\{\mathbf{x} \in \mathcal{X} : \frac{\sum_{y\in\mathcal{Y}} \mathbb{1}\left[y \neq f_{\boldsymbol{\theta}}(\mathbf{x})\right] \cdot \mathcal{N}(\mathbf{x}, y\boldsymbol{\mu}, \sigma^2 I)}{\sum_{y\in\mathcal{Y}} \mathbb{1}\left[y = f_{\boldsymbol{\theta}}(\mathbf{x})\right] \cdot \mathcal{N}(\mathbf{x}, y\boldsymbol{\mu}, \sigma^2 I)} > \gamma\right\}. \tag{35}$$

We are now able to state the optimal discriminator for this setting.

Definition 3 (Optimal discriminator for the logistic regression and the Gaussian model)**.** For any $0 < \gamma < \infty$ and $\mathbf{x} \in \mathcal{X}$, the optimal discriminator follows as:

$$D^\star(\mathbf{x}, \gamma) \triangleq \mathbb{1}\left[\sum_{y\in\mathcal{Y}} \mathbb{1}\left[y \neq f_{\boldsymbol{\theta}}(\mathbf{x})\right] \cdot \mathcal{N}(\mathbf{x}, y\boldsymbol{\mu}, \sigma^2 I) > \gamma \cdot \sum_{y\in\mathcal{Y}} \mathbb{1}\left[y = f_{\boldsymbol{\theta}}(\mathbf{x})\right] \cdot \mathcal{N}(\mathbf{x}, y\boldsymbol{\mu}, \sigma^2 I)\right]. \tag{36}$$

Since we cannot analytically evaluate Proposition 3.1, we proceed numerically in the next experiment.

B.2 Experiments

In this section, we will numerically evaluate Proposition 3.1 via empirical estimates of Type-I and Type-II errors in expressions (5). Note that unlike section 4, in this case all the involved distributions are known and hence it is also possible to compute the *true posterior distribution* $P_{Y|X}$.

We adopt the same notation as in section 3 for DOCTOR, i.e., D_α, and D_β according to according to expressions (14). $D^\star$, as in Definition 3, denotes the optimal discriminator.

B.2.1 Experimental setup and evaluation metrics

Dataset. We create a synthetic dataset that consists of 5000 data points drawn from $\mathcal{N}_0 \triangleq \mathcal{N}(\boldsymbol{\mu}_0, \sigma^2 I)$ and 5000 data points drawn from $\mathcal{N}_1 \triangleq \mathcal{N}(\boldsymbol{\mu}_1, \sigma^2 I)$, where $\boldsymbol{\mu}_0 = [-1 \ -1]$, $\boldsymbol{\mu}_1 = [1 \ 1]$. We consider two values for sigma, namely $\sigma = 2$ and $\sigma = 4$. These values produce two different distributions which will let us showcase the advantages of DOCTOR. To each data point $\mathbf{x}$ is assigned as class 0 or 1 depending on whether $\mathbf{x} \sim \mathcal{N}_0$ or $\mathbf{x} \sim \mathcal{N}_1$, respectively. The aforementioned dataset is divided into a training set, i.e. $\mathcal{D}_n = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ where $n = 6700$, and a testing set, i.e. $\mathcal{T}_m = \{(\mathbf{x}_{n+1}, y_{n+1}), \ldots, (\mathbf{x}_{n+m}, y_{n+m})\}$ where $m = 3300$.

Table 3: Accuracy on the test set: $f_{\boldsymbol{\theta}_i}$ for $i = 1, \ldots, 8$ represents the i-th model in $\mathcal{F}$, f_{avg} is the arithmetic mean of the accuracy over each $f_{\boldsymbol{\theta}_i} \in \mathcal{F}$. The value $f^\star_{avg}$ represents the accuracy Bayesian classifier averaged on the test set corresponding to the 8 splits. We show results for both standard deviations, namely $\sigma = 2$ and $\sigma = 4$.

CLASSIFIER	ACCURACY%	
	$\sigma = 2$	$\sigma = 4$
$f_{\boldsymbol{\theta}_1}$	82	65
$f_{\boldsymbol{\theta}_2}$	83	77
$f_{\boldsymbol{\theta}_3}$	82	77
$f_{\boldsymbol{\theta}_4}$	82	76
$f_{\boldsymbol{\theta}_5}$	83	76
$f_{\boldsymbol{\theta}_6}$	81	66
$f_{\boldsymbol{\theta}_7}$	82	76
$f_{\boldsymbol{\theta}_8}$	83	83
f_{avg}	82	74
$f^\star_{avg}$	83	78

Training configuration. We use a linear classifier, with one hidden layer, sigmoid activation function and binary cross entropy loss. The neural network is trained with gradient descent considering learning rate $r = 0.1$. Specifically, we train our network for 5 epochs. We randomly split our dataset 8 times, each time keeping n samples to train, and m to test. We consider the same model architecture (described above) for each split and we come up with 8 different binary discriminators $\mathcal{F} = \{f_{\boldsymbol{\theta}_1}, \ldots, f_{\boldsymbol{\theta}_8}\}$. Since in this example all the involved distributions are known, we compute the optimal predictor, i.e. the Bayes classifier, and we denote it with $f^\star$. The value $f^\star_{avg}$ reported in table 3, represents its accuracy averaged on the test set corresponding to the 8 splits.

Accuracy of trained networks. In table 3 the accuracy of $f^\star$ and the models in $\mathcal{F}$ on the test set.

Evaluation metric. We consider the same metric as in section 4.2.

B.2.2 Numerical evaluation of Proposition 3.1

To evaluate Proposition 3.1 we proceed in a Monte Carlo fashion by computing Type-I and Type-II errors for each of the network in $\mathcal{F}$ and then averaging over the results. Schematically, consider any $f_{\boldsymbol{\theta}_i} \in \mathcal{F}$ and $\gamma = 1$, we compute:

1. $\mathcal{A}_i \triangleq \mathcal{A}_i(1)$ as defined in eq. (35) and its complement $\mathcal{A}_i^c$.
2. For each classifier $f_{\boldsymbol{\theta}_i} \in \mathcal{F}$, $\mathcal{T}_{E=1;\boldsymbol{\theta}_i} \triangleq \{(\mathbf{x}, y) \in \mathcal{T}_m \mid y \neq f_{\boldsymbol{\theta}_i}(\mathbf{x})\}$ represents the set of mis-classified test samples, and $\mathcal{T}_{E=0;\boldsymbol{\theta}_i} \triangleq \{(\mathbf{x}, y) \in \mathcal{T}_m \mid y = f_{\boldsymbol{\theta}_i}(\mathbf{x})\}$ is the set of correctly classified test samples.
3. $\mathcal{FR}_i \triangleq \{(\mathbf{x}, y) \in \mathcal{T}_{E=0;\boldsymbol{\theta}_i} : \mathbf{x} \in \mathcal{A}_i\}$, $\mathcal{TR}_i \triangleq \{(\mathbf{x}, y) \in \mathcal{T}_{E=1;\boldsymbol{\theta}_i} : \mathbf{x} \in \mathcal{A}_i\}$, $\mathcal{FA}_i \triangleq \{(\mathbf{x}, y) \in \mathcal{T}_{E=1;\boldsymbol{\theta}_i} : \mathbf{x} \in \mathcal{A}_i^c\}$ and $\mathcal{TA}_i \triangleq \{(\mathbf{x}, y) \in \mathcal{T}_{E=0;\boldsymbol{\theta}_i} : \mathbf{x} \in \mathcal{A}_i^c\}$, i.e. the set of false rejections, true rejections, false acceptances and true acceptance, respectively.
4. $\epsilon_0(\mathcal{A}_i) \triangleq \frac{|\mathcal{FR}_i|}{|\mathcal{T}_{E=0;\boldsymbol{\theta}_i}|}$ and $\epsilon_1(\mathcal{A}_i^c) \triangleq \frac{|\mathcal{FA}_i|}{|\mathcal{T}_{E=1;\boldsymbol{\theta}_i}|}$, i.e. Type-I and Type-II errors.

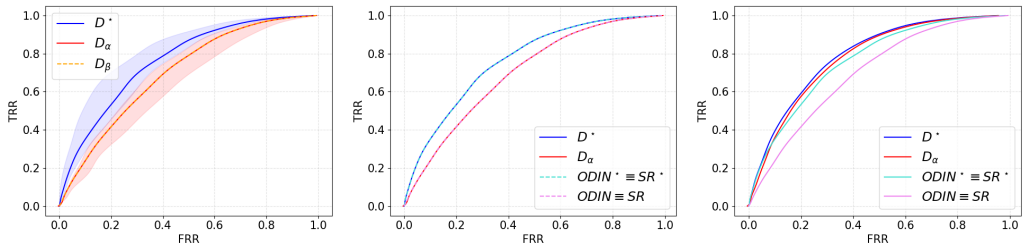
At the end of $|\mathcal{F}|$ iterations, we empirically estimate Type-I and Type-II errors of Proposition 3.1 as follows

$$\epsilon_0(\mathcal{A}) \approx \frac{1}{|\mathcal{F}|}\sum_{i=1}^{|\mathcal{F}|}\epsilon_0(\mathcal{A}_i) = 0.0607 \quad \text{and} \quad \epsilon_1(\mathcal{A}^c) \approx \frac{1}{|\mathcal{F}|}\sum_{i=1}^{|\mathcal{F}|}\epsilon_1(\mathcal{A}_i^c) = 0.7389.$$

B.2.3 FRR versus TRR

We present the experimental results obtained by running experiments similar to those described in section 4 considering the experimental setup in B.2.1 in TBB. In addition to the usual discriminators, we are going to consider the optimal discriminator $D^\star$, as in Definition 3.

DOCTOR: comparison between $D^\star$, D_α and D_β. Let us present the result obtained with DOCTOR showing how $D^\star$ (36) works compared to D_α and D_β in (14) when they have to decide whether to trust or not the decision made by a classifier. We test the discriminators on the dataset constructed as in B.2.1 by considering $\sigma = 2$. Let us analyze fig. 3a: we apply each discriminator to all the classifiers in $\mathcal{F}$. The obtained ROCs are represented by the colored areas. Inside each area the mean ROC is represented by the thick line. D_α and D_β reach same results as the colored areas and the thick lines are overlapped. For a given $\mathbf{x} \in \mathcal{X}$, we recall that $D^\star$ uses $\text{Pe}(\mathbf{x})$ (1) whilst D_α and D_β uses $1 - \widehat{\text{g}}(\mathbf{x})$ (11) and $\widehat{\text{Pe}}(\mathbf{x})$ (2), respectively. $D^\star$ always outperforms both D_α and D_β since it relies on the probability of classification error based on $P_{Y|X}$ while D_α and D_β use $P_{\widehat{Y}|X}$.



(a) $\sigma = 2$ (b) $\sigma = 2$ (c) $\sigma = 4$

Figure 3: ROC curves for $D^\star$, D_α and D_β, respectively. We denote by SR* the softmax response method based on $P_{Y|X}$. Since in this case $T = 1$ and $\epsilon = 0$, SR $\equiv$ ODIN as well as SR$^\star \equiv$ ODIN*. (a) We apply each discriminator to all the classifiers in $\mathcal{F}$. The obtained ROCs are represented by the colored areas. Inside each area the mean ROC is represented by the thick line. Orange and red areas completely overlap as well as the mean ROC. $D^\star$ always outperforms both D_α and D_β as expected. In (b) $D^\star$ and SR* overlap (as also D_α and SR), instead in (c) where $\sigma = 4$ and hence the distribution is smoother, SR discards useful information and indeed both $D^\star$ and D_α outperform SR.

Comparison between $D^\star$, D_α, ODIN and SR. We conclude this section by investigating how our competitors, namely ODIN and SR, work in this setting.

From now on, we will put ODIN $\equiv$ SR to mean that the two methods coincide (remember we set $T = 1$ and $\epsilon = 0$ for all the simulations). We show the results of the comparison in fig. 3: fig. 3b considers data points from $\mathcal{N}(y\boldsymbol{\mu}, 2^2 I)$ whilst fig. 3c consider data points from $\mathcal{N}(y\boldsymbol{\mu}, 4^2 I)$. If in fig. 3b we cannot see an advantage in using D_α in place of SR, the situation is totally different in fig. 3c, where $D^\star$ and D_α clearly outperform the competitors. We would like to recall that DOCTOR uses all the softmax output while SR only uses the maximum value of the softmax output. Therefore, when the underlying distribution p_{XY} is more smooth like in fig. 3c, SR discards useful information. As result, not only $D^\star$ outperforms SR* but even D_α does the same. This is more

Table 4: AUROCs: the values for D_α, D_β, SR, and ODIN correspond to the results for the thick lines in fig. 3. $D^\star$ and ODIN$^\star \equiv$ SR* are obtained using $P_{Y|X}$.

	AUROC %				
σ	$D^\star$	D_α	D_β	SR $\equiv$ ODIN	SR$^\star \equiv$ ODIN*
2	**76**	70	70	70	76
4	**79**	78	78	70	76

Table 5: AUROCs and FRR at 95% TRR obtained via D_α, D_β, ODIN, SR and MHLNB for CIFAR10 considering different size for $\Gamma_{D_\alpha \text{ or } D_\beta}$, $\Delta_{\text{ODIN or SR}}$ and Z_{MHLNB} in both TBB and PBB. The column INTERVAL SIZE represents the number of equidistant values considered in the sets defined in (37), (38), (39), (40) and in (41), respectively.

INTERVAL SIZE	METHOD	TBB		PBB	
		AUROC	FRR (95 % TRR)	AUROC	FRR (95 % TRR)
10	D_α	69.8	91.6	77.4	88.4
	D_β	50	69.7	79.8	86.2
	ODIN	75.7	89.3	81.4	85.4
	SR	75.7	89.3	-	-
	MHLNB	76.6	88.8	83.2	47.1
100	D_α	85.1	80.6	92.5	42.6
	D_β	61.8	63.4	94.1	13.8
	ODIN	88	73.5	91.5	49.9
	SR	88	73.5	-	-
	MHLNB	88.3	72.6	84.4	44.6
1000	D_α	91.3	53.1	94.7	13.8
	D_β	66.5	48.3	94.8	13.4
	ODIN	92.5	28.9	94	18.3
	SR	92.5	28.9	-	-
	MHLNB	92.2	35.3	84.4	44.5
10000	D_α	93.7	18.4	95.2	13.9
	D_β	68.5	18.6	94.8	13.4
	ODIN	93.9	18	94.2	18.4
	SR	93.9	18	-	-
	MHLNB	92.1	31	84.4	44.6

evident if we look to table 3, where for $\sigma = 4$ we notice an improvement in terms of AUROC from 70% to 78% when passing from SR to D_α.

C Supplementary Results of Section 4

C.1 Experimental environment

We run each experiment on a machine equipped with an Intel(R) Xeon(R) CPU E5-2623 v4, 2.60GHz clock frequency, and a GeForce GTX 1080 Ti GPU. The execution time for the execution the tests are the following (interval size 10000):

TBB. D_α: 12.5 s. D_β: 13.6 s. SR: 15.9 s. MHLNB: 15.9 s.

PBB: D_α: 13 s. D_β: 25.7 s. ODIN: 14.7 s. MHLNB: 32.22 s.

C.2 On the input pre-processing in DOCTOR

In the following we further study DOCTOR-specific input pre-processing techniques allowed under PBB. We focus on D_β since for D_α the reasoning is the same. Formally, let $\mathbf{x}_0 \in \mathcal{X}$ be a testing sample. We are looking for the minimum way to perturb the input such that the discriminator value at $\mathbf{x}_0$ is increased:

$$r^* = \min_{r \text{ s.t. } \|r\|_\infty \leq \epsilon} -\log\left(\frac{\widehat{\text{Pe}}(\mathbf{x}_0 + r)}{1 - \widehat{\text{Pe}}(\mathbf{x}_0 + r)}\right),$$

or equivalently, we are looking to the sample $\widetilde{\mathbf{x}}_0^\beta$ in the ϵ-ball around $\mathbf{x}_0$ which maximize the discriminator value at $\widetilde{\mathbf{x}}_0^\beta$:

$$\widetilde{\mathbf{x}}_0^\beta = \mathbf{x}_0 - \epsilon \times \text{sign}\left[-\nabla_{\mathbf{x}_0} \log\left(\frac{\widehat{\text{Pe}}(\mathbf{x}_0)}{1 - \widehat{\text{Pe}}(\mathbf{x}_0)}\right)\right].$$

Note that, because of eq. (1)

$$\begin{aligned}
-\log\left(\frac{\widehat{\text{Pe}}(\mathbf{x}_0)}{1 - \widehat{\text{Pe}}(\mathbf{x}_0)}\right) &= -\log\left(\frac{1 - P_{\widehat{Y}|X}(f_{\mathcal{D}_n}(\mathbf{x}_0)|\mathbf{x}_0)}{P_{\widehat{Y}|X}(f_{\mathcal{D}_n}(\mathbf{x}_0)|\mathbf{x}_0)}\right) \\
&= -\log(1 - P_{\widehat{Y}|X}(f_{\mathcal{D}_n}(\mathbf{x}_0)|\mathbf{x}_0)) + \log(P_{\widehat{Y}|X}(f_{\mathcal{D}_n}(\mathbf{x}_0)|\mathbf{x}_0)) \\
&= -\log(1 - P_{\widehat{Y}|X}(f_{\mathcal{D}_n}(\mathbf{x}_0)|\mathbf{x}_0)) - \log \text{SODIN}(\mathbf{x}_0).
\end{aligned}$$

C.3 On the effect the intervals considered for γ, δ and ζ have on the AUROC computation

Let us consider the AUROC as a performance measure for the discriminators. The computation of the AUROC of D_α, as well as those of ODIN and SR, heavily depend on the choice of the range

values for the decision region thresholds. In the following paragraph, we will discuss how we chose these ranges, namely $\gamma \in \Gamma_{D_\alpha \text{ or } D_\beta} \subseteq \mathbb{R}$, $\delta \in \Delta_{\text{ODIN or SR}} \subseteq [0,1]$ and $\zeta \in Z_{\text{MHLNB}} \subseteq \mathbb{R}$. In the experiments of section 4, we therefore proceed by fixing the aforementioned ranges as follows:

$$\Gamma_{D_\alpha} \triangleq \left[\min_{(\mathbf{x},y)\in\mathcal{T}_m} \frac{1-\widehat{\mathrm{g}}(\mathbf{x})}{\widehat{\mathrm{g}}(\mathbf{x})}, \max_{(\mathbf{x},y)\in\mathcal{T}_m} \frac{1-\widehat{\mathrm{g}}(\mathbf{x})}{\widehat{\mathrm{g}}(\mathbf{x})} \right], \tag{37}$$

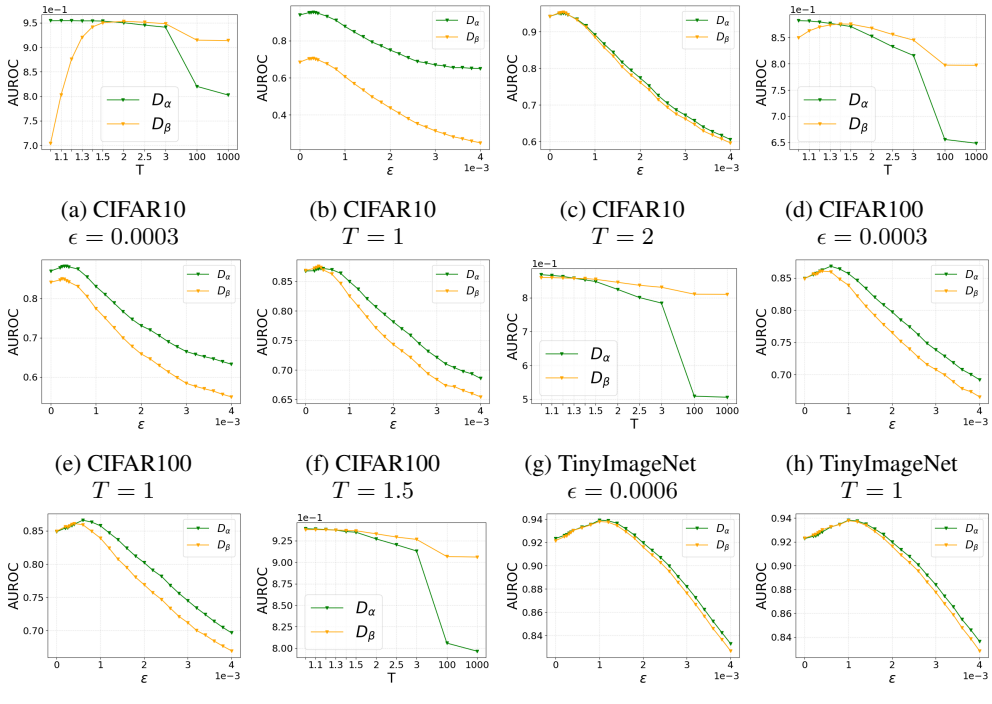
$$\Gamma_{D_\beta} \triangleq \left[\min_{(\mathbf{x},y)\in\mathcal{T}_m} \frac{\widehat{\mathrm{Pe}}(\mathbf{x})}{1-\widehat{\mathrm{Pe}}(\mathbf{x})}, \max_{(\mathbf{x},y)\in\mathcal{T}_m} \frac{\widehat{\mathrm{Pe}}(\mathbf{x})}{1-\widehat{\mathrm{Pe}}(\mathbf{x})} \right], \tag{38}$$

$$\Delta_{\text{ODIN}} \triangleq \left[\min_{(\mathbf{x},y)\in\mathcal{T}_m} \text{SODIN}(\mathbf{x}), \max_{(\mathbf{x},y)\in\mathcal{T}_m} \text{SODIN}(\mathbf{x}) \right], \tag{39}$$

$$\Delta_{\text{SR}} \triangleq \left[\min_{(\mathbf{x},y)\in\mathcal{T}_m} \text{SR}(\mathbf{x}), \max_{(\mathbf{x},y)\in\mathcal{T}_m} \text{SR}(\mathbf{x}) \right], \tag{40}$$

$$Z_{\text{MHLNB}} \triangleq \left[\min_{(\mathbf{x},y)\in\mathcal{T}_m} \text{M}(\mathbf{x}), \max_{(\mathbf{x},y)\in\mathcal{T}_m} \text{M}(\mathbf{x}) \right]. \tag{41}$$

Secondly, we fix the number of values to consider in $\Gamma_{D_\alpha \text{ or } D_\beta}$, $\Delta_{\text{ODIN or SR}}$ and Z_{MHLNB}: we test the AUROCs for CIFAR10 for different values of the size of $\Gamma_{D_\alpha \text{ or } D_\beta}$, $\Delta_{\text{ODIN or SR}}$ and Z_{MHLNB} in both TBB and PBB scenarios. The results are collected in table 5. Let us denote by I a generic interval between the ones of eq. (37), eq. (38), eq. (39), eq. (40) and eq. (41), throughout the experiments we set the size of I to $(\max I - \min I) * 10000$.



(a) CIFAR10 $\epsilon = 0.0003$ (b) CIFAR10 $T = 1$ (c) CIFAR10 $T = 2$ (d) CIFAR100 $\epsilon = 0.0003$

(e) CIFAR100 $T = 1$ (f) CIFAR100 $T = 1.5$ (g) TinyImageNet $\epsilon = 0.0006$ (h) TinyImageNet $T = 1$

(i) TinyImageNet $T = 1.1$ (j) SVHN $\epsilon = 0.001$ (k) SVHN $T = 1$ (l) SVHN $T = 1.2$

Figure 4: Comparison of AUROCs obtained via D_α (in green) and via D_β (in orange) for different values of T and ϵ.

C.4 Additional plots and results

In the next sections, we show graphically the set of results obtained from the experiments in section 4.3. We first specify the range of values for the parameters T and ϵ considered throughout the experiments. For temperature scaling, T is selected among

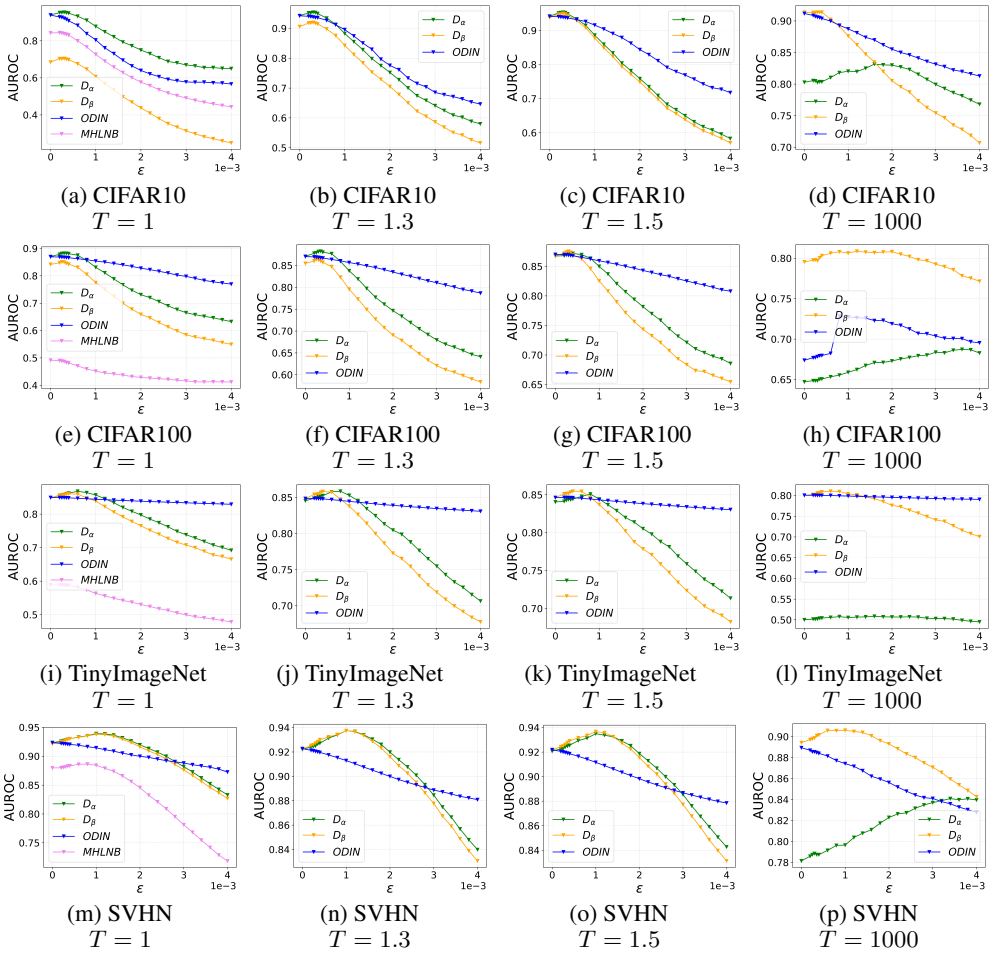
$\{1,\ 1.1,\ 1.2,\ 1.3,\ 1.4,\ 1.5,\ 2,\ 2.5,\ 3,\ 100,\ 1000\}$, whilst for input pre-processing, ϵ is selected among $\{0,\ .0002,\ .00025, .0003,\ .00035,\ .0004, .0006,\ .0008,\ .001, .0012,\ .0014,\ .0016,\ .0018, .002,\ .0022, .0024,\ .0026,\ .0028, .003,\ .0032,\ .0034, .0036,\ .0038,\ .004\}$.

C.4.1 Comparison D_α and D_β

We include the plots for DOCTOR*: comparison between D_α and D_β* (section 4.3). In fig. 4a, fig. 4d, fig. 4g and fig. 4j, we set ϵ at its best value which is found to coincide in the case of D_α and D_β. In fig. 4b, fig. 4e, fig. 4h and fig. 4k we do the opposite and we set T to its best value w.r.t. D_α whilst in fig. 4c, fig. 4f, fig. 4i and fig. 4l, the value of T is chosen w.r.t. the best value for D_β.

C.4.2 Comparison D_α, D_β, ODIN and MHLNB

We conclude by showing in appendix C.4.2 the test results obtained by varying T and ϵ in PBB for all the methods. We present 4 groups of plots (one for each image dataset) and in each plot we pick T from $\{1, 1.3, 1.5, 1000\}$ (the values selected for D_α, D_β, ODIN and MHLNB table 1) and we let ϵ vary.



(a) CIFAR10 $T = 1$ (b) CIFAR10 $T = 1.3$ (c) CIFAR10 $T = 1.5$ (d) CIFAR10 $T = 1000$

(e) CIFAR100 $T = 1$ (f) CIFAR100 $T = 1.3$ (g) CIFAR100 $T = 1.5$ (h) CIFAR100 $T = 1000$

(i) TinyImageNet $T = 1$ (j) TinyImageNet $T = 1.3$ (k) TinyImageNet $T = 1.5$ (l) TinyImageNet $T = 1000$

(m) SVHN $T = 1$ (n) SVHN $T = 1.3$ (o) SVHN $T = 1.5$ (p) SVHN $T = 1000$

C.4.3 Misclassification detection in presence of out-of-distribution samples

We include in table 6 the results of all the simulations carried out for detecting misclassification detection in presence of out-of-distribution samples. The experimental setting is reported in section 4.2.

C.5 DOCTOR for pure OOD detection

It is worth emphasizing that DOCTOR is not targeting OOD detection, which is a rather different problem from the one investigated in this paper. So we did not optimize an ad-hoc input perturbation for DOCTOR within the OOD detection setup, i.e. we kept the same input perturbation proposed for the misclassification detection task. The baseline results reported in table 7 show that DOCTOR is competitive for OOD detection as well since it can reach similar scores or even outperform the baseline (e.g., the simulations with LSUN (CROP) show an improvement of the results of 3.3% in terms of FRR $\%$). We indicate the methods together with their parameter setting. ODIN_{OOD} denotes the same parameter setting as in [23].

C.5.1 DOCTOR in presence of OOD samples that are similar to in-distribution ones

We tested DOCTOR in pure OOD setting, considering CIFAR100 as in-distribution and CIFAR10 as out-distribution. The results below show that DOCTOR optimized as in the following paper outperforms ODIN (optimized as described in [23]) and ENERGY. This is particularly promising as it shows that DOCTOR, without performing any training and without been particularly optimized for OOD detection, can perform well on a wider variety of problems.

C.6 Some observations on the white-box scenario (WB)

It is worth clarifying the results in Table 9 to motivate the performance obtained using the Mahalanobis-based discriminator (MHLNB - WB) for the misclassification detection problem and the issues it raises. First of all, we emphasize that given a network and an input sample DOCTOR only needs to access the logits output of the network in order to perform the detection. On the contrary, the detector based on Mahalanobis distance consists of 3 steps:

- Estimation of the class mean and covariance matrix;
- Features extraction according to the Mahalanobis score function;
- Aggregation of the scores obtained layer by layer in order to obtain a decision a rule for the discriminator.

Clearly, the Mahalanobis distance-based method requires additional samples compared to DOCTOR. Although estimating the mean and the covariance matrix is possible by exploiting samples from the benchmark tra.et (e.g. CIFAR10, CIFAR100, ...), this method still needs additional (different from training) samples for learning the linear regressor intended to distinguish between correctly (positive) and incorrectly (negative) classified samples. In order to generate the negative samples, we consider the use of adversarial examples generated through Projected Gradient Descent Attack (magnitude of the perturbation 0.0031), which does not assume any knowledge about the test set.

Table 6: In PBB we set $\epsilon_\alpha = 0.00035$ and $T_\alpha = 1$, $\epsilon_\beta = 0.00035$ and $T_\beta = 1.5$, $\epsilon_{\text{ODIN}} = 0$ and $T_{\text{ODIN}} = 1.3$. By ODIN_{ood}, we mean ODIN with the parameter setting as in [23]. Since we proceed in a Monte Carlo fashion, the results are reported in terms of *mean / standard deviation*. In TBB for by ODIN we report the results of SR, since both methods coincide when $T = 1$ and $\epsilon = 0$.

DATASET (IN)	DATASET (OUT)	SCENARIO	AUROC %				FRR % (95 % TRR)			
			D_α	D_β	ODIN	ODIN_{OOD}	D_α	D_β	ODIN	ODIN_{OOD}
CIFAR10 ♣	iSUN	PBB	**95.4** / 0.1	95.1 / 0.1	94.6 / 0.1	89.6 / 0	14 / 0.5	**13.5** / 0.4	17.2 / 0.3	38.9 / 0
		TBB	**94.6** / 0	69.3 / 0.1	94.5 / 0.1	-	**17.7** / 0.1	**17.7** / 0.1	**17.7** / 0	-
	LSUN (CROP)	PBB	**95.5** / 0.1	95.1 / 0	94.7 / 0	92.6 / 0	13.1 / 0.5	**13** / 0.2	17.3 / 0	31.9 / 0.1
		TBB	**94.4** / 0.1	69.2 / 0.1	**94.4** / 0	-	**17.6** / 0.2	**17.6** / 0.2	17.7 / 0.2	-
	LSUN (RESIZE)	PBB	**95.4** / 0.1	95.1 / 0	94.8 / 0	89.6 / 0	13.4 / 0.6	**13.2** / 0.3	17 / 0.3	38.9 / 0
		TBB	**94.6** / 0.1	69.3 / 0.1	94.5 / 0.1	-	**17.8** / 0.1	**17.8** / 0.1	**17.8** / 0.1	-
	TINY (CROP)	PBB	**95.4** / 0	95.1 / 0.1	94.7 / 0	89.6 / 0	13.4 / 0.4	**13** / 0.2	17.2 / 0.3	38.9 / 0
		TBB	**94.6** / 0	69.4 / 0.1	**94.6** / 0	-	**17.8** / 0.1	**17.8** / 0.1	**17.8** / 0.1	-
	TINY (RES)	PBB	**95.2** / 0.1	94.9 / 0	94.6 / 0.1	89.6 / 0	**14** / 0.4	**14** / 0.5	17.8 / 0.4	38.9 / 0
		TBB	**94.4** / 0.1	69.2 / 0	**94.4** / 0	-	**17.8** / 0.1	**17.8** / 0.1	**17.8** / 0.1	-
CIFAR100 ♣	iSUN	PBB	**86.5** / 0.2	85.8 / 0	85.6 / 0.2	79 / 0.1	**45.3** / 1	46.1 / 0.5	46.8 / 1	65.9 / 0.4
		TBB	**85.6** / 0.1	82.7 / 0.1	85.5 / 0.1	-	46.9 / 0.4	**46.8** / 0.4	**46.8** / 0.4	-
	LSUN (CROP)	PBB	**89.1** / 0	88.5 / 0.1	88 / 0.1	80.6 / 0	**35.6** / 0.4	35.7 / 0.2	39.9 / 0.3	65.1 / 0
		TBB	**87.9** / 0.1	84.9 / 0.1	87.7 / 0.1	-	**39.8** / 0.6	**39.8** / 0.6	**39.8** / 0.6	-
	LSUN (RESIZE)	PBB	**86.8** / 0.1	86.2 / 0.1	86 / 0.1	79.1 / 0.1	**44.4** / 0.9	**44.4** / 0.6	45.3 / 0.3	65.4 / 0.3
		TBB	**85.8** / 0.1	82.9 / 0.1	85.7 / 0.1	-	45.9 / 0.5	**45.8** / 0.5	**45.8** / 0.5	-
	TINY (CROP)	PBB	**88.4** / 0.1	87.8 / 0.1	87.6 / 0.1	81.8 / 0.1	38.2 / 0.4	**37.8** / 0.9	40.6 / 0.5	63.4 / 0.1
		TBB	**87.2** / 0.1	84.2 / 0.1	87 / 0.1	-	**42** / 0.6	**42** / 0.6	**42** / 0.6	-
	TINY (RES)	PBB	**86.8** / 0.1	86.3 / 0.1	85.9 / 0.1	79.2 / 0.1	44 / 0.1	**43.6** / 0.2	45.9 / 1.2	65.8 / 0.3
		TBB	**85.9** / 0.2	83 / 0.2	85.8 / 0.2	85.8 / 0.2	**45.7** / 1.3	**45.7** / 1.3	**45.7** / 1.3	-
CIFAR10 ♢	iSUN	PBB	**95.5** / 0.1	95.3 / 0.1	94.9 / 0.1	91.5 / 0	14.4 / 0.6	**13.4** / 0.2	16.8 / 0.5	34/ 0.1
		TBB	**95** / 0	69.6 / 0	94.9 / 0.1	-	**16.4** / 0.2	**16.4** / 0.2	**16.4** / 0.2	-
	LSUN (CROP)	PBB	**95.8** / 0.1	95.5 / 0.1	95 / 0.1	93.9 / 0.1	**12.4** / 0.2	12.6 / 0.1	16.1 / 0.4	24.8 / 0.1
		TBB	**94.8** / 0.1	69.6 / 0.1	**94.8** / 0.1	-	16.7 / 0.4	16.8 / 0.4	**16.6** / 0.4	-
	LSUN (RESIZE)	PBB	**95.8** / 0	95.6 / 0	95.2 / 0	91.6 / 0	**12.9** / 0.5	**12.9** / 0.3	15.8 / 0.2	33.9 / 0
		TBB	**95** / 0	69.7 / 0.1	**95** / 0.1	-	**16.4** / 0.2	**16.4** / 0.3	**16.4** / 0.2	-
	TINY (CROP)	PBB	**95.8** / 0.1	95.5 / 0.1	95.2 / 0.1	91.5 / 0	**12.8** / 0.7	12.9 / 0.5	16 / 0	33.9 / 0
		TBB	**95** / 0.2	69.8 / 0.1	**95** / 0.1	-	**16.4** / 0.2	16.5 / 0.2	**16.4** / 0.2	-
	TINY (RES)	PBB	**95.4** / 0.1	95 / 0.1	94.8 / 0.1	91.4 / 0	15 / 0.1	**14.8** / 0.7	17 / 0.5	34.5 / 0.9
		TBB	**94.6** / 0.2	69.3 / 0.2	**94.6** / 0.2	-	18.1 / 1	18.1 / 1.1	**18** / 1	-
CIFAR100 ♢	iSUN	PBB	**84.8** / 0.1	84.4 / 0.2	84.6 / 0.1	80.8 / 0.2	53.6 / 1	**51.2** / 0.2	51.3 / 0.1	63.5 / 0.3
		TBB	**84.1** / 0.1	81.2 / 0.1	84 / 0.1	-	**52.5** / 0.5	**52.5** / 0.5	**52.5** / 0.5	-
	LSUN (CROP)	PBB	**89.9** / 0.1	89.6 / 0	89 / 0	84.1 / 0	**35.2** / 0.7	35.4 / 0.2	39.3 / 0.1	62.2 / 0
		TBB	**88.7** / 0.1	85.7 / 0	88.5 / 0.1	-	**38.8** / 0.5	**38.8** / 0.5	**38.8** / 0.4	-
	LSUN (RESIZE)	PBB	**85.3** / 0.3	85.1 / 0.2	84.9 / 0.1	81.1 / 0	51.6 / 0.9	**48.8** / 1	49.2 / 0.7	63.3 / 0.1
		TBB	**84.6** / 0.2	81.8 / 0.2	**84.6** / 0.1	-	**50.6** / 0.8	50.7 / 0.8	**50.6** / 0.8	-
	TINY (CROP)	PBB	**88.2** / 0	88.1 / 0.2	87.7 / 0.1	84.8 / 0.1	41.2 / 0.3	**40.2** / 0.6	42.3 / 0.4	59/ 0.2
		TBB	**87.7** / 0.1	84.7 / 0.1	87.5 / 0.1	-	**41.8** / 0.5	**41.8** / 0.5	**41.8** / 0.5	-
	TINY (RES)	PBB	**85.4** / 0.2	84.8 / 0.2	85.1 / 0.3	81.2 / 0.1	51.8 / 1.6	52 / 0.8	**50.4** / 0.9	63.3 / 0.2
		TBB	**84.8** / 0.1	81.9 / 0.1	84.7 / 0.1	-	**51.4** / 0.5	**51.4** / 0.5	**51.4** / 0.5	-
CIFAR10 ♠	iSUN	PBB	**95.6** / 0.1	**95.6** / 0	95.4 / 0	93.5 / 0	15.1 / 0.1	**13.6** / 0.5	16.1 / 0.2	30.6 / 0.4
		TBB	**95.4** / 0.1	70 / 0.1	95.2 / 0.1	-	16.1 / 0.4	**16** / 0.5	**16** / 0.4	-
	LSUN (CROP)	PBB	**96.1** / 0.1	95.9 / 0.1	95.5 / 0.2	95.2 / 0.1	12.6 / 0.5	**12.4** / 0.3	15.3 / 0.7	20.8 / 0.4
		TBB	**95.2** / 0.1	70 / 0.1	**95.2** / 0.1	-	15.8 / 0.7	15.8 / 0.7	**15.7** / 0.7	-
	LSUN (RESIZE)	PBB	**96** / 0	95.8 / 0	95.7 / 0	93.6 / 0	13.2 / 0.5	**13** / 0.2	15.2 / 0.4	30.3 / 0.4
		TBB	**95.5** / 0.1	70.2 / 0.1	**95.5**/ 0.1	-	15.2 / 0.5	15.2 / 0.5	**15.1** / 0.5	-
	TINY (CROP)	PBB	**96** / 0.1	95.9 / 0.1	95.7 / 0	93.6 / 0	13.5 / 0.9	**12.7** / 0.4	15.2 / 0.4	30.3 / 0.4
		TBB	95.5 / 0.1	70.3 / 0	**95.6** / 0	-	15.1 / 0.2	**15** / 0.3	**15** / 0.2	-
	TINY (RES)	PBB	**95.5** / 0.1	95.2 / 0.1	95.1 / 0.1	93.2	**14.7** / 0.3	14.8 / 0.5	17.1 / 0.4	31/ 0
		TBB	**94.9** / 0.1	69.7 / 0.1	**94.9** / 0.1	-	16.8 / 0.3	16.9 / 0.2	**16.7** / 0.2	-
CIFAR100 ♠	iSUN	PBB	**83.3** / 0.1	83.1 / 0.1	83 / 0.2	82.6 / 0.2	57.8 / 0.3	57.1 / 1	**56.8** / 0.8	60/ 0.4
		TBB	**82.6** / 0.2	79.7 / 0.2	82.5 / 0.2	-	**58.3** / 1	58.4 / 1.1	58.4 / 1	-
	LSUN (CROP)	PBB	90.6 / 0	**90.7** / 0	89.9 / 0.1	87.5 / 0	35.9 / 0.2	**34.6** / 0.2	38.5 / 0.4	56.1 / 0.2
		TBB	**89.4** / 0.1	86.2 / 0	89 / 0	-	**39.4** / 0.1	**39.4** / 0.1	**39.4** / 0.1	-
	LSUN (RESIZE)	PBB	83.6 / 0.2	**83.8** / 0.1	83.6 / 0.2	83.2 / 0.1	55.8 / 0.4	54.2 / 0.7	**54.1** / 0.6	59.6 / 0.8
		TBB	**83.2** / 0.1	80.4 / 0.1	**83.2** / 0.1	-	**55** / 0.6	**55** / 0.7	**55** / 0.6	-
	TINY (CROP)	PBB	88.3 / 0.1	**88.5** / 0.1	88.1 / 0.1	87.7 / 0.1	43.2 / 0.5	**41.5** / 0.7	42.9 / 0.4	54.3 / 0.1
		TBB	**87.8** / 0	84.7 / 0.1	87.5 / 0.1	-	**43.7** / 0.2	**43.7** / 0.2	**43.7** / 0.2	-
	TINY (RES)	PBB	83.8 / 0.1	83.8 / 0.1	**83.9** / 0.2	83/ 0.2	57.9 / 0.5	56.6 / 0.9	**55.6** / 1	61/ 0.6
		TBB	**83.6** / 0.1	80.7 / 0.1	83.5 / 0.1	-	**55.5** / 0.8	**55.5** / 0.8	**55.5** / 0.8	-

Table 7: DOCTOR for pure OOD detection. We set : $\epsilon_\alpha = 0$ and $T_\alpha = 15$, $\epsilon_\beta = 0$ and $T_\beta = 1000$, as in [23] for ODIN_{OOD}. The baseline results reported below show that DOCTOR is competitive for OOD detection as well since it can reach similar scores or even outperform the baseline.

DATASET-IN	**DATASET-OUT**	**AUROC** %			**FRR** % **(95** % **TRR)**		
		D_α	D_β	ODIN_{OOD}	D_α	D_β	ODIN_{OOD}
CIFAR10	iSUN	98.1	97.9	**98.8**	8	9.1	**6.3**
	TINY (RES)	97.6	97.3	**98.5**	9.9	11.2	**7.2**
	LSUN (CROP)	**98.6**	98.2	98.2	**5.4**	6.9	8.7
	TINY (CROP)	98.9	98.5	**99.1**	4.6	6.4	**4.3**

Table 8: Comparison of D_α with ENERGY and ODIN (parameter setting as in [23]) when OOD samples are similar to in-distribution samples.

DATASET-IN	**DATASET-OUT**	**METHODS**	**AUROC** %	**FRR** % **(95** % **TRR)**
CIFAR100	CIFAR10	D_α (PBB)	76.8	64.2
		ENERGY	73.3	76.4
		ODIN (OOD)	70.5	79.5

Table 9: Comparison of MHLNB (WB) and D_α (PBB).

DATASET-IN	**METHODS**	**AUROC** %	**FRR** % **(95** % **TRR)**
CIFAR10	D_α (PBB)	95.2	13.9
	MHLNB (WB)	49.5	97.3
CIFAR100	D_α (PBB)	88.2	35.7
	MHLNB (WB)	51.6	94.9