DOCTOR: A Simple Method for Detecting Misclassification Errors

Anonymous Author(s)
Affiliation
Address
email

Abstract

Deep neural networks (DNNs) have shown to perform very well on large scale object recognition problems and lead to a 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 which 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 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 [16]. Developing methods and tools to make these algorithms 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 [6, 7, 8, 10, 24, 33, 34]. 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, 11, 20, 22, 31] 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 [11]. In general, most of these works consider *white-box* scenarios,

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

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 [22] which involves the use of temperature scaling and the addition of small perturbations to input samples. A related solution is introduced in [5] 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 [13, 21], which consists in calculating how much the observed out-distribution sample deviate from the in-distribution ones but assuming the latter are given.

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.

1. 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. (9) allows us to devise a feasible implementation of it, based on the softmax probability (Proposition 3.2).

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.

3. From the experimental point of view, we show that DOCTOR outperforms comparable state-of-the-art methods (e.g., ODIN [22], softmax response [5] and Mahalanobis distance [21]) 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).

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 [8, 18, 19]. Furthermore, models often tend to be overconfident in their decision even when their predictions fail [10, 16]. 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 [22], the authors propose a method which increases the peakiness of the softmax output by adding perturbations to the input samples and applying temperature scaling [8, 12, 27] 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 [13, 21] tackles OOD detection by using *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 to reduce the error rate by abstaining from prediction when in doubt, while keeping the number of correctly classified samples as high as possible [5, 6]. 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 [5], an interesting rejection function which compares the maximum soft-probability is introduced, called *softmax response*, with a pre-determined threshold to decide whether to accept or reject the class prediction given by the model.

2 Main Definitions and Preliminaries

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 $\mathcal{Y} = \{1, \ldots, C\}$ 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 $\mathcal{D}_n = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\} \sim p_{XY}$ 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, while $P_{Y|X}$ is the true (unknown) probability. Indeed, in many practical scenarios it is often the case that $P_{\widehat{Y}|X}$ does not perfectly match $P_{Y|X}$ and still $f_{\mathcal{D}_n} \approx f^\star$ (see [8]).

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 $\widehat{E}(\mathbf{x}) \triangleq \mathbb{1}\left[\widehat{Y} \neq f_{\mathcal{D}_n}(\mathbf{x})\right]$ 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:

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

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

Obviously, $\mathrm{Pe}^\star(\mathbf{x}) \leq \mathrm{Pe}(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{X}$, where $\mathrm{Pe}^\star(\mathbf{x})$ corresponds to the minimum error of the Bayes classifier: $\mathrm{Pe}^\star(\mathbf{x}) = 1 - P_{Y|X}\left(f^\star(\mathbf{x})|\mathbf{x}\right)$. 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 $\widehat{\mathrm{Pe}}(\mathbf{x})$ provides a valuable candidate to infer the unknown error variable $E(\mathbf{x})$, it is easy to check that

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

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 $\mathrm{Pe}(\mathbf{x}) : \mathcal{X} \mapsto [0, 1]$ plays a central role in the characterization of the optimal discriminator. However, $\mathrm{Pe}(\mathbf{x})$ 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 [30].

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 those 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,

$$\epsilon_0(\mathcal{A}) + \epsilon_1(\mathcal{A}^c) \geq 1 - \left\| p_{X|E=1} - p_{X|E=0} \right\|_{\mathrm{TV}} = 1 - \frac{1}{2} \int_{\mathcal{X}} |p_{X|E=1}(\mathbf{x}) - p_{X|E=0}(\mathbf{x})| d\mathbf{x}. \tag{6}$$

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:

$$2 \Pr\left\{ D(\mathbf{X}) \neq E(\mathbf{X}) \right\} \geq 1 - \left\| p_{X|E=1} - p_{X|E=0} \right\|_{\mathrm{TV}}. \tag{7}$$

Equality is achieved by using the optimal decision region.

Expressions (6) and (7) 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:

$$\mathcal{A}(\gamma) = \left\{ \mathbf{x} \in \mathcal{X} : P_{E|X}(1|\mathbf{x}) P_E(0) > \gamma \cdot \left(1 - P_{E|X}(1|\mathbf{x})\right) P_E(1) \right\}. \tag{8}$$

From (8), 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:

$$\mathcal{A}(\gamma') = \left\{ \mathbf{x} \in \mathcal{X} : \frac{\mathrm{Pe}(\mathbf{x})}{1 - \mathrm{Pe}(\mathbf{x})} > \gamma' \right\} = \left\{ \mathbf{x} \in \mathcal{X} : \mathrm{Pe}(\mathbf{x}) > \frac{\gamma'}{(\gamma' + 1)} \right\}, \tag{9}$$

where $\gamma' \triangleq \gamma \cdot \frac{P_E(1)}{P_E(0)}$ and $0 < \gamma' < \infty$. According to (9) 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 (9). First, we state the following:

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

$$1 - \widehat{\mathrm{g}}(\mathbf{x}) \triangleq \sum_{y \in \mathcal{Y}} P_{\widehat{Y}|X}(y|\mathbf{x}) \Pr\left(\widehat{Y} \neq y | \mathbf{x}\right) = 1 - \sum_{y \in \mathcal{Y}} P^2_{\widehat{Y}|X}(y|\mathbf{x}), \tag{10}$$

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,

$$(1 - \sqrt{\widehat{\mathrm{g}}(\mathbf{x})}) - \Delta(\mathbf{x}) \leq \mathrm{Pe}(\mathbf{x}) \leq (1 - \widehat{\mathrm{g}}(\mathbf{x})) + \Delta(\mathbf{x}), \tag{11}$$

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).

Discussion. It is worth emphasizing that expressions in (11) 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,

$$\Pr\big(\Delta(\mathbf{X}) \geq \varepsilon(\eta)\big) \leq \eta \quad \text{with} \quad \varepsilon(\eta) = 2\sqrt{2\mathbb{E}_{\mathbf{X}Y}\big[-\log P_{\widehat{Y}|X}(Y|\mathbf{X})\big]/\eta}, \tag{12}$$

for any $\eta > 0$, where $\mathbb{E}_{\mathbf{X}Y}\big[-\log P_{\widehat{Y}|X}(Y|\mathbf{X})\big]$ in (12) 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, (10) turns out to be related to the uncertainty of the classifier via the quadratic Rényi entropy [29]: $-\log_2\big(\widehat{\mathrm{g}}(\mathbf{x})\big) = 2H_2(\widehat{Y}|\mathbf{x}) \leq 2H(\widehat{Y}|\mathbf{x})$, where the latter is the Shannon entropy, i.e., the self-uncertainty of the classifier.

From theory to a practical discriminator. Our previous discussion suggests that $\widehat{\mathrm{Pe}}(\mathbf{x})$ in (2) may be a valuable candidate to approximate $\mathrm{Pe}(\mathbf{x})$ in the definition of the optimal discriminator (9). On the other hand, Proposition 3.2 suggests that $1 - \widehat{\mathrm{g}}(\mathbf{x})$ 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:

$$D_\alpha(\mathbf{x}, \gamma) \triangleq \mathbb{1}\left[1 - \widehat{\mathrm{g}}(\mathbf{x}) > \gamma \cdot \widehat{\mathrm{g}}(\mathbf{x})\right], \qquad D_\beta(\mathbf{x}, \gamma) \triangleq \mathbb{1}\left[\widehat{\mathrm{Pe}}(\mathbf{x}) > \gamma \cdot (1 - \widehat{\mathrm{Pe}}(\mathbf{x}))\right]. \tag{13}$$

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 [22], softmax response (SR) [5] and Mahalanobis distance (MHLNB) [21]. 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:

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

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 [22] and in [21], respectively.

4.1 Review of related methods

PBB. We compare DOCTOR (using input pre-processing and temperature scaling) with ODIN and MHLNB. ODIN [22] comprises temperature scaling and input pre-processing via perturbation.

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

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 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). Unlike all the methods considered in this paper, MHLNB [21] requires the knowledge of the training set $\mathcal{D}_n$ which the pre-trained network was trained on to compute its *empirical class mean* $\widehat{\mu}_c = \frac{1}{n_c} \sum_{i:\, y_i = c} f(\widetilde{\mathbf{x}}_i)$ for each class c and its *empirical covariance* $\widehat{\Sigma} = \frac{1}{n} \sum_{c \in \mathcal{Y}} \sum_{\forall i:\, y_i = c} (f(\widetilde{\mathbf{x}}_i) - \widehat{\mu}_c)(f(\widetilde{\mathbf{x}}_i) - \widehat{\mu}_c)^\top$, where n_c denotes the number of training samples with label c and $f(\widetilde{\mathbf{x}})$ the vector of logits. As MHLNB directly uses the vector of logits, it does not comprise temperature scaling. Formally, given an input sample $\mathbf{x}$:

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

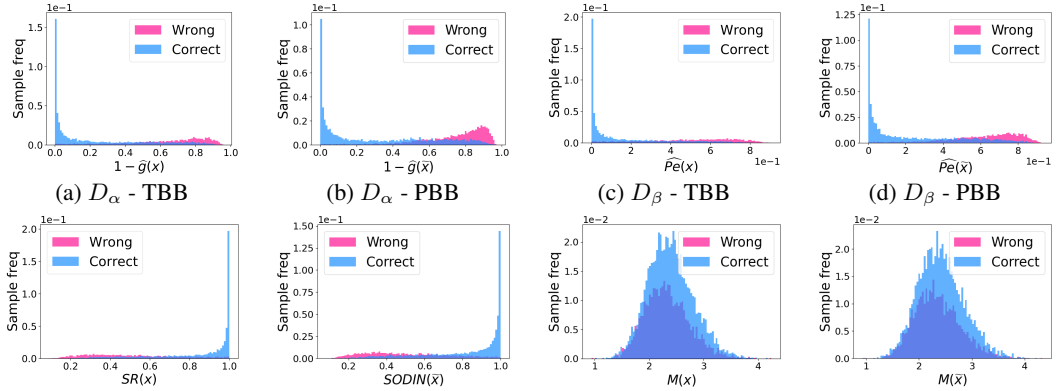
as mentioned above, $\widetilde{\mathbf{x}}$ represents a magnitude ϵ perturbation of the original $\mathbf{x}$; $\zeta \in \mathbb{R}_+$ is the threshold value; in indicates the acceptance decision while out indicates the rejection decision.

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 [5] 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 fig. 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 between 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 fig. 1a and fig. 1b for D_α, fig. 1c and fig. 1d for D_β, fig. 1e for SR, fig. 1f for ODIN, fig. 1g and fig. 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 optimally chooses acceptance and rejection regions.

From fig. 1g and fig. 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 fig. 1e and fig. 1f, respectively, does not look significantly different from the one related to D_α and D_β in TBB (eq. (13)). However, the discrimination between samples becomes evident in PBB. This is shown in fig. 1d for D_β (eq. (13)) and even more in fig. 1b for D_α (eq. (13)) where, quite clearly,



(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 (10); (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).

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.

Datasets. We run experiments on both image and textual datasets. We use CIFAR10 and CIFAR100 [17], TinyImageNet [15] and SVHN [25] as image datasets; IMDb [23], AmazonFashion and AmazonSoftware [26] as textual datasets. Note that, for all the aforementioned datasets, we consider only the test set since we rely on pre-trained models.

Pre-trained networks. Along the same lines of [22], we use the pre-trained DenseNet models [14] for CIFAR10, CIFAR100 and SVHN. In addition, we use a pre-trained ResNet model [9] for TinyImageNet, and BERT [3, 32] for the Amazon datasets and IMDb. The accuracy achieved by the aforementioned networks on the test sets is showed in table 1. 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 $\mathbf{x}_i$ among C possible classes and m denotes the size of the testing set. Then, considering $j \in \{\alpha, \beta\}$: $\mathcal{FR}_j(\gamma) = \{(\mathbf{x}, y) \in \mathcal{T}_m : y = f_{\mathcal{D}_n}(\mathbf{x}),\ D_j(\mathbf{x}, \gamma) = 1\}$, $\mathcal{TR}_j(\gamma) = \{(\mathbf{x}, y) \in \mathcal{T}_m : y \neq f_{\mathcal{D}_n}(\mathbf{x}),\ D_j(\mathbf{x}, \gamma) = 1\}$, $\mathcal{FA}_j(\gamma) = \{(\mathbf{x}, y) \in \mathcal{T}_m : y \neq f_{\mathcal{D}_n}(\mathbf{x}),\ D_j(\mathbf{x}, \gamma) = 0\}$ and $\mathcal{TA}_j(\gamma) = \{(\mathbf{x}, y) \in \mathcal{T}_m : y = f_{\mathcal{D}_n}(\mathbf{x}),\ D_j(\mathbf{x}, \gamma) = 0\}$.

We measure the performance of the test in terms of:

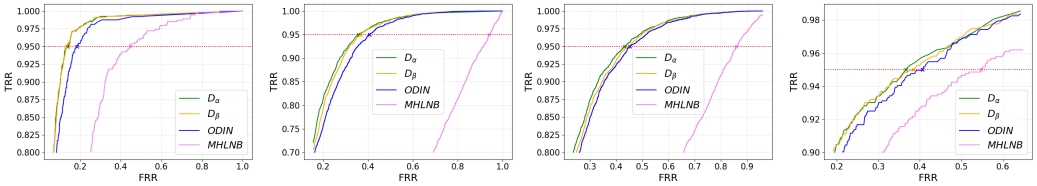
- **FRR** versus **TRR**. The false rejection rate (FRR) represents the probability that a hit is rejected, while the true rejection rate (TRR) is the probability that a miss is rejected.
- **AUROC**. The area under the *Receiver Operating Characteristic curve* (ROC) [2] depicts the relationship between 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

DOCTOR: comparison between D_α and D_β. We compare the discriminators D_α and D_β introduced in (13) 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. For space reason we refer to Supplementary material (Appendix C.4.1) for the plots. In contrast to what has been noted with temperature scaling, the perturbation magnitude ϵ has the same effect on both D_α and D_β: in all datasets we noticed that the best results are obtained for the same epsilon values of D_α and D_β.

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 $\epsilon_{\text{ODIN}} = 0$) and MHLNB ($T_{\text{MHLNB}} = 1$ and $\epsilon_{\text{MHLNB}} = 0.0002$). The red dashed line marks the 95% threshold of TRR.

Comparison in PBB. We compare DOCTOR with ODIN and MHLNB. 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 Supplementary material (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. According to our framework no validation samples are available; consequently, in order to be consistent across the dataset, we only report the experimental settings and values for which, on average, we obtain good results for all the considered domains: $\epsilon_\alpha = 0.00035$ and $T_\alpha = 1$, $\epsilon_\beta = 0.00035$ and $T_\beta = 1.5$, $\epsilon_{\text{ODIN}} = 0$ and $T_{\text{ODIN}} = 1.3$, $\epsilon_{\text{MHLNB}} = 0.0002$ and $T_{\text{MHLNB}} = 1$. When ODIN is applied to the problem of deciding whether to trust or not the decision made by a classifier, we do not keep the same parameter setting as in [22] ($T = 1000$ and ϵ is set to 0.0014 for CIFAR10, 0.002 for CIFAR100, 0.001 for TinyImageNet and 0.0002 for SVHN) since our experiments show that this setup causes deterioration in performance (see plots in Supplementary material (Appendix C.4.2)) .

DOCTOR's performance compared to ODIN's and MHLNB's, are collected in table 1 and in fig. 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 since the training set on which the network was trained is not available. In Supplementary material we give simulations on how the range of interval for γ, δ and ζ can affect the results (Appendix C.3).

Misclassification detection in presence of OOD samples. We evaluate DOCTOR's performance in the task of detecting misclassifications under 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 ($\clubsuit$), three ($\diamondsuit$) or two ($\spadesuit$) is OOD. The details of the simulations and the considered dataset, as well as the collection of all the experimental results, are relegated to Supplementary material (Appendix C.4.3). In table 2 we report an extract of the results for the PBB scenario: DOCTOR is able to achieve (and most of the time outperform) 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 : $\epsilon_\alpha = 0.00035$ and $T_\alpha = 1$, $\epsilon_\beta = 0.00035$ and $T_\beta = 1.5$, $\epsilon_{\text{ODIN}} = 0$ and $T_{\text{ODIN}} = 1.3$, $\epsilon_{\text{MHLNB}} = 0.0002$ and $T_{\text{MHLNB}} = 1$. In TBB for ODIN we report same results as in SR, since both methods coincide when $T = 1$ 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
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
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

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
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; as in [22] for ODIN_{OOD}. Since we proceed in a Monte Carlo fashion, the results are in terms of *mean / standard deviation*.

DATASET-IN	DATASET-OUT	AUROC %				FRR % (95 % TRR)			
		D_α	D_β	ODIN	ODIN_{OOD}	D_α	D_β	ODIN	ODIN_{OOD}
CIFAR10 ♣	iSUN	**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
	TINY (RES)	**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
CIFAR10 ♢	iSUN	**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
	TINY (RES)	**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
CIFAR10 ♠	iSUN	**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
	TINY (RES)	**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

5 Summary and Concluding Remarks

In this paper, we studied a simple but 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. Our result advances the state-of-the-art work, and the main takeaway is that DOCTOR can be applied to both totally and partially black-box setups. In the former scenario, the only information comes from the samples to be tested and the classifier soft-probabilities, in the latter, gradient-propagation to perform input pre-processing is allowed. In practice, this is particularly important since the information about the model's architecture may be undisclosed for security reason when dealing with sensitive data. DOCTOR exploits all the information conveyed by the softmax output, which results in equal or better performance with respect to the other methods: particularly promising are the results for the partially black-box scenario, where we observe a reduction up to 4% in terms of predictions incorrectly rejected with respect to the ones for the totally black-box scenario. Moreover, DOCTOR does not require training data and, thanks to its flexibility, it can be easily deployed in real-world scenarios. Although the proposed framework seeks to protect users and to discourage blind trust in potentially untrustworthy automated decisions, we are aware that practitioners who lack the necessary knowledge and skills may incur misinterpretation of the results obtainable with the proposed method. 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.

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] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

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

[13] 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.

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

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

[16] 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.

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

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

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

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

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

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

[30] 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.

[31] 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.

[32] 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.

[33] C. Xing, S. Ö. Arik, Z. Zhang, and T. Pfister. Based learning from errors for hangds2019corrion. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

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

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Please, refer to lines 330-331.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We briefly discuss societal impacts or our work in lines 344-346. Furthermore, to the best of our knowledge, the current application of our framework and the utilized data do not raise any ethical concerns.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] The full body of theoretical statements and results is reported in section 3.
   (b) Did you include complete proofs of all theoretical results? [Yes] The full body of proofs for the theoretical results is reported in appendix A.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] A link to the code repository is provided in line 183. Instructions to run the aforementioned code and reproduce the results is provided within the repository as a README.md file.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A] Models' training phase not needed in the proposed framework.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A] Only for experiments including randomness in the data we repeated several runs on different instances and expressed the results in terms of mean and standard deviation. In addition, in the code repository (cf. 3.a) we provide the datasets we used for the experiments or a link to them.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Cf. appendix C.1.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A] We only used openly available code which we carefully referred throughout the paper.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] A link to the code repository and related instructions is provided in line 183.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We utilized standard benchmark openly available datasets.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]