Utilizing Class Separation Distance for the Evaluation of Corruption Robustness of Machine Learning Classifiers

Georg Siedel[1,2,*], Silvia Vock[1], Andrey Morozov[2] and Stefan Voß[1]

[1]Federal Institute for Occupational Safety and Health (BAuA) Germany
[2]University of Stuttgart, Germany

Abstract

Robustness is a fundamental pillar of Machine Learning (ML) classifiers, substantially determining their reliability. Methods for assessing classifier robustness are therefore essential. In this work, we address the challenge of evaluating corruption robustness in a way that allows comparability and interpretability on a given dataset. We propose a test data augmentation method that uses a robustness distance ε derived from the datasets minimal class separation distance. The resulting MSCR (mean statistical corruption robustness) metric allows a dataset-specific comparison of different classifiers with respect to their corruption robustness. The MSCR value is interpretable, as it represents the classifiers avoidable loss of accuracy due to statistical corruptions. On 2D and image data, we show that the metric reflects different levels of classifier robustness. Furthermore, we observe unexpected optima in classifiers robust accuracy through training and testing classifiers with different levels of noise. While researchers have frequently reported on a significant tradeoff on accuracy when training robust models, we strengthen the view that a tradeoff between accuracy and corruption robustness is not inherent. Our results indicate that robustness training through simple data augmentation can already slightly improve accuracy.

Keywords

corruption robustness, classifier, class separation, metric, accuracy-robustness-tradeoff

1. Introduction

ML functions are deployed to an increasing extent over various industries including machinery engineering. Within the European domestic market, machinery products are subject to regulation of the Machinery directive, which demands a risk assessment[1].

Risk assessment includes risk estimation and evaluation, where risk is defined as a combination of probability and severity of a hazardous event. Therefore, once ML functions are deployed in machinery products, where their failure may lead to a hazardous event, being able to quantify the probability and severity of their failures becomes mandatory. However, there still exists a gap between the regulative and normative requirements for safety critical software and the existing methods to assess ML safety [1].

This work targets ML classifiers, the failures of which are misclassifications. Our focus is on the evaluation of failure probability specifically, not on failure severity. We address one specific failure mode of ML classifiers: Corrupted or perturbed data inputs that cause a change of the output to a misclassification. The property of a classifier resistant to any such input corruptions is called robustness[2]. A classifier is a function that assigns a class to any d-dimensional input $x \in R^d$. Classifier g is robust at a point x within a distance $\varepsilon > 0$, if $g(x) = g(x^{'})$ holds for all perturbed points $x^{'}$ that satisfy $dist(x - x^{'}) \leq \varepsilon$ [2, 3]. The *dist*-function can e.g. be an L_p-norm distance, while ε can be defined based on physical observations of e.g. which perturbations are imperceptible for humans.

Robustness is considered a desirable property since intuitively, a slightly perturbed input (e.g. an imperceptibly changed image) should not lead to a classifier changing its corresponding prediction. In essence, a robustness requirement demands that within a certain input parameter space around x, all points $x^{'}$ have to share the same class. This way, a robustness requirement adds additional information on how the classifier should behave near ground truth data points. Authors therefore argue the importance of robustness, being a fundamental pillar of reliability [4] and quality [5] of ML models.

However, popular robustness training methods show significantly lowered test accuracy compared to standard training, which has lead to some authors discussing an inherent, i.e. inevitable tradeoff between accuracy and robustness (see Section 2.2).

Two types of robustness need to be clearly distinguished [5, 6, 7]: adversarial robustness and corruption robustness.

The IJCAI-ECAI-22 Workshop on Artificial Intelligence Safety (AISafety 2022), July 24-25, 2022, Vienna, Austria
*Corresponding author.
siedel.georg@baua.bund.de (G. Siedel)
https://github.com/georgsiedel/minimal-separation-corruption-robustness (G. Siedel)
© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

[1]Machinery Directive, Directive 2006/42/EC of the European Parliament and of the Council of 17 May 2006.

[2]Robustness includes resistance to any corruption-caused class change, which may not be a failure mode when the original point was already misclassified (cf. footnote 4).

CEUR-WS.org/Vol-3215/20.pdf

Adversarial inputs are perturbed data deliberately optimized to fool a classifier into changing its output class. Corruption robustness (sometimes: statistical robustness) describes a model's output stability not against such worst-case, but against statistically distributed input corruptions. The two types of robustness require different training methods and are differently hard to achieve depending on the data dimension [6]. In practice, training a model for one of the two robustness types only shows limited or selective improvement for the other type [7, 8, 9].

In the field of research towards ML robustness, most of the attention has been given to adversarial attack and defense methods. However, from the perspective of machinery safety and risk assessment, adversarial robustness is mainly a security concern and therefore not in the scope of this article. [10] argue that instead of adversarial robustness evaluation, a corruption robustness evaluation is often more applicable to obtain a real-world robustness measure and it can be used to estimate a probability of failure on potentially perturbed inputs for the overall system.

Contribution: In this paper, we investigate corruption robustness using data augmentation for testing and training[3]. Our key contributions are twofold:

- We propose the "*MSCR*" metric to evaluate and compare classifiers corruption robustness. The approach is independent of prior knowledge about corruption distances, but utilizes properties of the underlying dataset, giving the metric a distinct interpretable meaning. We show experimentally, that the metric captures different levels of classifier corruption robustness.
- We evaluate the tradeoff between accuracy and robustness from the perspective of corruption robustness and present arguments against the tradeoff being inherent.

After giving an overview of related work, we present our approach for the MSCR metric in section 3.1. We then test our approach on simple 2D as well as image data with the setup described in section 3.2. We present and discuss the results in sections 4 and 5.

2. Related Work

2.1. Measuring corruption robustness

Corruption robustness of classifiers can be numerically evaluated by testing the ratio of correctly/incorrectly classified inputs from a corrupted test dataset. This ratio is called robust accuracy/error, in contrast to the ratio of correct classification on original test data ("clean accuracy/error"). Robust accuracy represents a combined measure for accuracy and robustness[4]. A useful way to obtain a measure of robustness only is by subtracting robust accuracy/error and clean accuracy/error [8, 11].

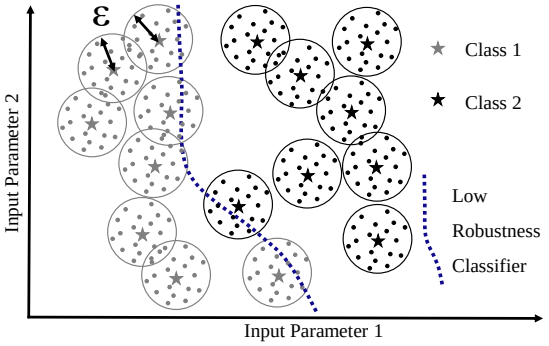


Figure 1: A robustness requirement (here: L_2-norm balls with maximum distance ε) assigned to the data points (stars) of a 2D binary dataset (2 input parameters, 2 classes). The shown classifier is not robust, since its dotted decision boundary violates the robustness requirement. To evaluate this, additional points (dots) are augmented within ε of each original point. On those points, the robust accuracy of the classifier is measured – for this classifier, some errors arise.

In most cases, the corrupted test dataset is derived from an original test dataset through data augmentation. One or multiple corruptions out of some distribution are added to every original data points. Figure 1 explains this procedure of data augmentation with corruptions (dots) being added to a test dataset (stars) with 2 parameters and 2 classes. It illustrates how a 100% accurate but non-robust classifier achieves lower robust accuracy on the augmented data points.

The corruption distribution can be defined e.g. based on physical observations. For the example of image data, [8, 12] add corruptions like brightness, blur and contrast, while [13, 14] use special weather or sensor corruptions. [8] created robustness benchmarks for the most popular image datasets based on such physical corruptions.

Corruption distributions can also be defined without physical representations by adding e.g. Gaussian-, salt-and-pepper-, or uniformly distributed noise of certain magnitude to the inputs [8, 11, 14, 10]. Figure 1 exemplary demonstrates uniformly distributed noise within L_2-norm distance ε (in 2D, L_2-norm is a circle) of the data points.

With PROVEN, [15] propose a framework that uses statistical data augmentation to estimate bounds on ad-

[3]Code available on Github, see front page.

[4]The term astuteness can be used for robust accuracy to differentiate the term from robustness, see [3]. Throughout this work, we use the popular term robustness to describe our metric for consistency with works like [8] and [10].

versarial robustness of a model, essentially combining the evaluation of both adversarial and corruption robustness.

[4] take a robustness evaluation approach different from measuring robust accuracy. The authors augment the entire input space with uniformly distributed data points, independent of a test dataset. They divide the input space into cells, the size of which is based on the r-separation distance described in [3] and in section 2.2. This way, they can assign a conflict free ground truth class to each cell and evaluate the misclassification ratio on all added data points. The approach allows for statistical testing of the entire input space, but does not scale well to high dimensions.

An analytical way of measuring the robustness of a classifier is through describing the characteristics of its decision boundary. One possibility is to estimate the local Lipschitzness, i.e. a tightened continuity property of models in proximity to data points. To the best of our knowledge however, Lipschitzness has only been used to investigate adversarial, not corruption robustness [2, 3].

Both the measure in [4] and Lipschitzness values lack distinct interpretability in terms of what the calculated value represents exactly.

2.2. The Accuracy-Robustness-Tradeoff

Significant effort has recently been put into increasing classifier robustness, commonly targeting adversarial robustness, e.g. in [9, 16, 17, 18, 19]. All these methods cause a significant drop in clean accuracy.

[11, 20] and [10] observe a clear tradeoff between corruption robustness and accuracy for different training methods using data augmentation. The two former works then propose specialized training methods for mitigating parts of this tradeoff on the popular image datasets CIFAR-10 and ImageNet.

Based on such research, [19] and [21] discuss a tradeoff between accuracy and robustness, while [22] even argue that the cause for this tradeoff is inherent, i.e. inevitable. A counterargument is presented by [3], who argue that accuracy and robustness are not necessarily at odds as long as data points from different classes are separated far enough from each other (see section 3). The authors measure this "r-separation" between different classes on various image datasets and find it to be high enough for classifiers to be both accurate and robust for typical perturbation distances.

3. Method

Our robustness evaluation approach is based on this same idea by [3], who measure the distance 2r for a dataset, which is the minimal distance between any two points of different classes (2r in Figure 2). The authors argue that

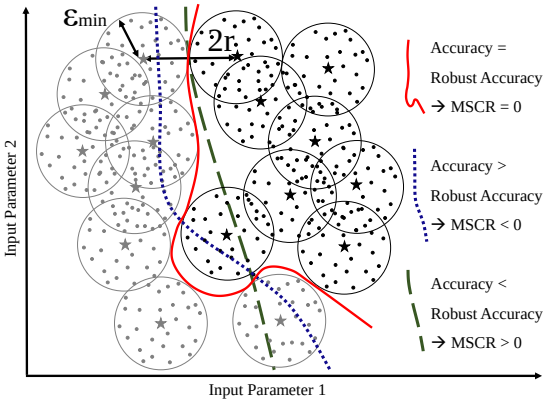


Figure 2: The MSCR concept, demonstrated on 2D test data. Data augmentation is carried out like in Figure 1. The distance (ε_{min}) is determined by the minimal distance ($2r$) of original points from different classes (black and grey). This way, augmented points of different classes are still separated and classifiers can be both accurate and robust. The decision boundaries of 3 hypothetical classifiers are shown to demonstrate different levels of robustness and their resulting MSCR value.

a classifier can be both robust and accurate as long as

$$\varepsilon \leq r \tag{1}$$

holds, where ε is the corruption distance for which robustness is evaluated and r is half this minimal class separation distance. We adopt this notation and set $\varepsilon_{min} = r$ as our corner case corruption distance (see Figure 2). The value ε_{min} is not related to any prior physical knowledge of e.g. which corruptions are imperceptible, but is specific for the given dataset, i.e. it is based on the fundamental property of minimal class separation. Accordingly, we call our metric "Minimal Separation Corruption Robustness" (MSCR).

3.1. MSCR metric

To measure corruption robustness, we carry out data augmentation on the test data with uniformly distributed corruptions, generated by a random sampling algorithm, similar to the method shown by [10]. In contrast to [10], we set the upper bound of the distance ε_{test}, within which the augmented noise is distributed, to ε_{min}, as required in Equation 1 (see Figure 2 for an illustration). We measure robust accuracy on the augmented data, which corresponds to a combination of clean accuracy and corruption robustness. However, we want to quantify robustness independent of clean accuracy for comparability, so we subtract the clean accuracy (Acc_{clean}) from the robust accuracy on ε_{min}-augmented test data ($Acc_{rob-\varepsilon min}$) and normalize by the clean accuracy:

$$MSCR = (Acc_{rob-\varepsilon min} - Acc_{clean})/Acc_{clean} \tag{2}$$

According to [3], a classifier can in principle be robust on such augmented noise of magnitude ε_{min} while maintaining accuracy. This can be seen from Figure 2, where the circles of radius ε_{min} in which data is augmented, never overlap for different classes. We use an identical radius ε_{min} for all classes, assuming that the separation of data points from the classifiers decision boundary is equally important for all classes. For this noise level ε_{min}, any non-robust behavior is theoretically avoidable, since a classifiers decision boundary can separate the classes even with augmented data, as long as the ML algorithm is capable of learning the exact function. The MSCR metric therefore measures the (relative) win or loss in accuracy when testing on such noisy data that any loss is just about avoidable. Figure 2 illustrates the impact of the proposed metric using three corner cases:

- $MSCR = 0$, $Acc_{rob-\varepsilon min} = Acc_{clean}$, solid line in Figure 2: A classifier that is as robust as possible for the given class separation of the dataset. It not only correctly classifies the original data points, but also all augmented data points.
- $MSCR < 0$, $Acc_{rob-\varepsilon min} < Acc_{clean}$, dotted line in Figure 2: A classifier that is not perfectly robust. It correctly classifies all original data points, but misclassifies a number of augmented data points due to low robustness.
- $MSCR > 0$, $Acc_{rob-\varepsilon min} > Acc_{clean}$, dashed line in Figure 2: A classifier misclassifies some original data points, but correctly classifies some of their augmentations. Especially for classifiers that trained to be very robust, we expect this result to be possible.

Algorithm 1 shows the MSCR calculation procedure. In step 1, different distance functions (e.g. L_∞-norm) can be applied. We account for randomness in the data splitting, model training and data augmentation procedures by carrying out multiple runs of the same experiment and reporting average values and 95%-confidence intervals over all runs. The reasonable number of augmented points k per original data point varies depending on the dataset (see section 3.2). Within the respective for-loop, variable *models* runs through the list of all classifier models to be compared, while *r* counts up to (the overall number of) *runs*.

Algorithm 1: MSCR calculation

Data: classification dataset
$\{X(x_1, ..., x_n), Y(y_1, ..., y_n)\}$
Parameters: $models = \{model_1, ..., model_m\}$,
$r = \{1, ..., runs\}$, k, $\varepsilon_{test} = \{0, \varepsilon_{min}\}$
Output: $\overline{MSCR} = \{\overline{MSCR_1}, ..., \overline{MSCR_m}\}$

1 $\varepsilon_{min} = (\min_{x_{i\in n}, x_{j\in n}} \{dist(x_j - x_i) | y_i \neq y_j\})/2$
2 **for** *models* **do**
3 **for** *r* **do**
4 Train $model_m$
5 Test model with original test data ($\varepsilon_{test} = 0$) → return Acc_{clean}
6 For every test data point: Uniform random sample k points within $dist(\varepsilon_{min})$ and augment the test data
7 Test model with data from step 6 → return $Acc_{rob-\varepsilon min}$
8 $MSCR_r = (Acc_{rob-\varepsilon min} - Acc_{clean})/Acc_{clean}$
9 $\overline{MSCR_m} = (\sum_{r=1}^{runs} MSCR_r)/runs$

3.2. Experimental details

Additionally to test data augmentation, we train multiple models on datasets augmented with different corruption distances ε_{train}. Increasing a model's ε_{train} should lead to a growing MSCR value, as it is expected that the model robustness grows. This way, we evaluate the trend of the MSCR value for models with different corruption robustness levels. Also, on test data corrupted with large ε_{test}, models trained with $\varepsilon_{train} = \varepsilon_{test}$ are expected to perform best [10].

As demonstrated in Figure 2, corruption levels below ε_{min} theoretically allow a classifier to be robust while not losing test accuracy. We investigate this theoretical claim by [3] additionally to the MSCR metric by evaluating changes in robust accuracy when augmenting multiple corruption levels ε_{test} to the test dataset. In contrast to the work of [10], we extensively evaluate more corruption levels below, around and including ε_{min} specifically. In contrast to the work of [11] and [20], we use simple uniformly distributed data augmentation with a fixed upper bound of noise for the entire dataset instead of Gaussian noise. This allows us the comparison of the noise levels with the class separation distances. It shall be noted however that in contrast to Gaussian noise, where density decreases with distance, uniform noise does not reflect the higher uncertainty in a class assignment when the distance from a ground truth data point increases. Even though our data augmentation method is simple, we still expect to find counterexamples for the accuracy-robustness-tradeoff, based solely on the class-separation theory. We believe that the case of finding such counterexamples with less advanced methods than e.g. [11] represents even more credible evidence for the argument of [3] against an inherent accuracy-robustness-tradeoff.

We carry out the experiments on 3 binary class 2D datasets as were used and provided by [4]. For clarity, we only report results with L_∞-corruptions on one of those datasets, which is shown in Figure 3 and features 4674 data points. Experiments with the other 2D datasets and L_2-corruptions exhibit similar fundamental results,

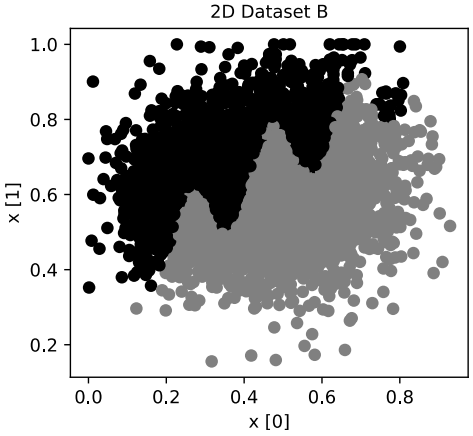


Figure 3: Data points in the binary class 2D dataset.

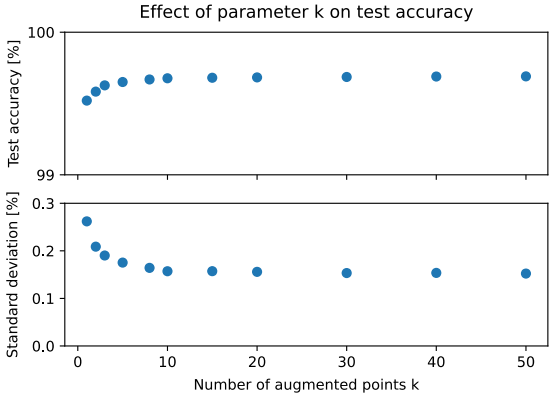


Figure 4: Effect of hyperparameter k on robust accuracy and its deviation. 2D dataset, $\varepsilon_{train}, \varepsilon_{test} = 0.001$.

which can also be found in our Github repository (see frontpage). The two input parameters $x[0]$ and $x[1]$ are normalized to the interval $[0, 1]$. For classification, we use a random forest (RF) algorithm with 100 trees. We also compare this classifier with a 1-nearest-neighbor model, which is known to be inherently robust, since it classifies based on distance to the 1 nearest data point.

We choose $k = 10$ augmented data points per original data point, as we found higher numbers of k not significantly improving the resulting robust accuracy and its standard deviation. This effect of different values for the hyperparameter k is displayed in Figure 4. In order to achieve statistically representative results, we evaluate how the average test accuracy converges over multiple runs and accordingly choose 1200 runs.

The experiments are additionally run in a more applied image classification setting using benchmark dataset CIFAR-10. We adopt the classifier architecture from [10], using a 28-10 wide residual network with SGD optimizer, 0.3 dropout rate, training batch size 32 and 30 epochs with a 3-step decreasing learning rate. All pixel values are normalized to $[0, 1]$ and random horizontal flips and random crops with 4px padding are used for training generalization. For CIFAR-10 we choose $k = 1$, since [10] report one augmented point to be sufficient. We suspect that this is due to the multiple epochs of the training process, which allows to train the model on multiple augmentations per training data point. We choose 20 runs due to computational feasibility of all training procedures. Table 1 shows the minimal class separation distances $2r$ and the corresponding ε_{min} values, measured in L_∞-distance for both datasets. For intuition, the CIFAR-10 ε_{min} value translates to a maximum color grade change of $27/255$ on all pixels. Higher values for $2r$ are to be expected for image data, since L_∞-norm evaluates the maximum distance in any of the 3072 dimensions of CIFAR-10 input data.

Table 1
Minimal L_∞ class separation and corresponding ε_{min}

Dataset	$2r(L_\infty)$	ε_{min}
2D dataset	0.008026	0.004013
CIFAR-10 (train and test set)	0.211765	0.105882

4. Results

Table 2 displays the matrix of test accuracies for the 2D dataset for different values of both ε_{train} (representing different models, along columns) and ε_{test} (along rows). The bold values highlight the best model for every level of test noise. As can be seen, the optima of the accuracy do not actually match with the matrix diagonal, where training and test noise are equal (highlighted in light grey). Instead, when testing with lower noise levels and even with clean test data, the model trained on $\varepsilon_{train} = 0.007$ performs best. The maximum overall accuracy is achieved with a model trained on $\varepsilon_{train} = 0.007$ that is also tested on $\varepsilon_{test} = 0.001$ corruptions. For higher noise levels, the optimum robust accuracies are achieved with $\varepsilon_{train} \leq \varepsilon_{test}$, displaying the opposite trend compared to low noise levels.

The results on CIFAR-10 in Table 3 show a similar trend, albeit less pronounced. For low noise levels, training with $\varepsilon_{train} = 0.01$ appears to be optimal for clean accuracy. The maximum overall accuracy is achieved with $\varepsilon_{train} = 0.02$ and $\varepsilon_{test} = 0.01$. For higher levels of test noise, similarly to 2D data, it appears beneficial to use $\varepsilon_{train} \leq \varepsilon_{test}$. In contrast to the 2D data, where the optimum ε_{train} for $\varepsilon_{test} = 0$ is higher than the ε_{min} value, for CIFAR-10 it is ~10 times lower than ε_{min}. The optimum $\varepsilon_{train} = 0.01$ translates to a $2.5/255$ color grade corruption for every pixel.

For both datasets it is visible from the last rows of

Table 2
Clean accuracies (first row) and robust accuracies in percentage plus the MSCR value (last row) for various models (columns) ± the 95% confidence intervals. Models are trained and tested with different levels of L_∞-noise (ε_{train} along columns, ε_{test} along rows). Bold accuracies: Best model accuracy for every noise level. Bold MSCR value: Highest MSCR value, i.e. highest model robustness. Last row color scale: Highlights the constant increase of MSCR with increasing ε_{train}. Light grey accuracies: Model trained and tested on the same noise level ($\varepsilon_{train} = \varepsilon_{test}$). Dark grey accuracies: Maximum overall accuracy.

2D Dataset	ε_{train}								
ε_{test}	**0**	**0.001**	**0.002**	ε_{min}	**0.007**	**0.01**	**0.015**	**0.02**	**0.03**
0	99.531±0.014	99.652±0.011	99.699±0.011	99.748±0.010	**99.784**±0.009	99.769±0.010	99.504±0.016	98.990±0.028	97.347±0.060
0.001	99.515±0.013	99.640±0.011	99.689±0.010	99.746±0.009	**99.785**±0.009	99.775±0.009	99.524±0.014	99.017±0.025	97.380±0.057
0.002	99.495±0.013	99.607±0.011	99.660±0.010	99.732±0.009	**99.777**±0.008	99.768±0.009	99.528±0.013	99.026±0.024	97.411±0.055
ε_{min}	99.405±0.013	99.525±0.011	99.583±0.010	99.669±0.009	**99.729**±0.008	99.716±0.008	99.486±0.012	99.005±0.022	97.435±0.052
0.007	99.167±0.014	99.287±0.012	99.360±0.011	99.461±0.010	**99.536**±0.009	99.535±0.010	99.319±0.013	98.871±0.022	97.380±0.049
0.01	98.782±0.017	98.899±0.015	98.977±0.014	99.083±0.014	99.175±0.013	**99.191**±0.014	99.014±0.017	98.615±0.024	97.238±0.047
0.015	97.871±0.025	97.979±0.025	98.044±0.025	98.134±0.025	98.222±0.025	**98.265**±0.026	98.197±0.029	97.921±0.033	96.810±0.049
0.02	96.771±0.036	96.847±0.037	96.896±0.037	96.966±0.037	97.040±0.038	97.092±0.038	**97.105**±0.040	96.962±0.043	96.198±0.053
0.03	94.397±0.058	94.423±0.059	94.456±0.059	94.500±0.060	94.547±0.061	94.593±0.061	94.668±0.061	**94.698**±0.062	94.448±0.066
MSCR	-0.126±0.007	-0.127±0.006	-0.116±0.006	-0.080±0.005	-0.055±0.005	-0.053±0.006	-0.018±0.010	0.015±0.015	**0.090**±0.024

(a) 2D Dataset

CIFAR-10 Dataset	ε_{train}							
ε_{test}	**0**	**0.01**	**0.02**	**0.03**	**0.05**	**0.07**	ε_{min}	**0.15**
0	91.681±0.304	**91.932**±0.318	91.917±0.417	91.311±0.470	90.428±0.427	88.645±0.454	86.051±0.800	81.989±0.855
0.01	91.453±0.338	91.900±0.311	**91.964±0.408**	91.351±0.474	90.472±0.421	88.678±0.463	86.085±0.798	81.995±0.858
0.02	90.675±0.429	91.527±0.338	**91.868**±0.428	91.459±0.442	90.577±0.420	88.817±0.457	86.158±0.763	82.082±0.844
0.03	89.181±0.595	91.606±0.385	**91.606**±0.462	91.479±0.422	90.690±0.403	88.983±0.421	86.306±0.766	82.165±0.838
0.05	84.062±1.033	89.273±0.611	89.273±0.570	90.530±0.483	**90.832**±0.346	89.513±0.362	86.745±0.737	82.540±0.802
0.07	76.706±1.396	84.086±0.999	84.086±0.831	87.303±0.672	**90.065**±0.324	89.907±0.332	87.322±0.621	83.051±0.789
ε_{min}	59.261±1.868	67.534±1.814	67.534±1.695	74.137±1.367	83.784±0.714	**88.260**±0.374	88.194±0.418	84.181±0.666
0.15	37.458±2.034	42.765±2.071	42.765±2.286	49.685±1.991	65.285±1.842	77.257±1.083	**86.130**±0.433	85.352±0.511
MSCR	-35.362±0.020	-33.977±0.020	-26.527±0.018	-18.808±0.014	-7.347±0.009	-0.434±0.006	2.490±0.006	**2.674**±0.003

(b) CIFAR-10 Dataset

Table 2 and 3, that the MSCR value steadily increases with higher levels of training noise ε_{train}. For both datasets, the MSCR increases from negative values on less robust trained models to zero and even positive values for more robust trained models.

For CIFAR-10, the MSCR values are overall much larger than for the 2D data. This effect correlates with the ε_{min} noise level, which is about 26 times larger in absolute values.

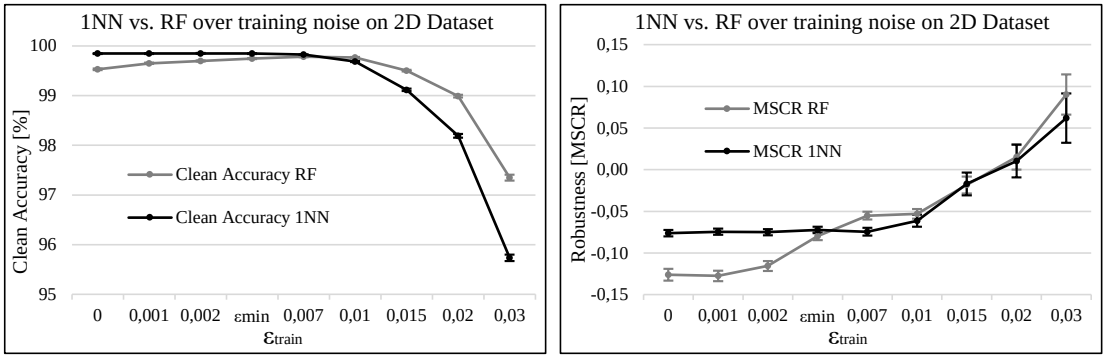
Figure 5 shows a comparison on the 2D dataset between the 1NN model and the RF model with regards to clean accuracy (Fig. 5a) and MSCR (Fig. 5b). Both models are trained on the various ε_{train} values. While for the RF model, both metrics increase with increasing training noise up to the optimum of $\varepsilon_{train} = 0.007$, the 1NN model shows constant (and superior) metrics up to this training noise. This illustrates the inherent robustness of the 1NN model. The comparison also shows that this inherent robustness is indeed advantageous regarding accuracy on our dataset.

Figures 6a (2D dataset) and 6b (CIFAR-10) display the accuracy-robustness-tradeoff for the models trained with different ε_{train} by contrasting MSCR versus clean accuracy values. Both Figures in principle show a tradeoff curve. However, it is visible that for $\varepsilon_{train} \leq 0.007$ on 2D data and $\varepsilon_{train} \leq 0.01$ on CIFAR-10, both clean accuracy and robustness increase compared to the baseline model with $\varepsilon_{train} = 0$. The tradeoff is overcome for these models (arguably also for $\varepsilon_{train} = 0.01$ for 2D data and $\varepsilon_{train} = 0.02$ for CIFAR-10).

5. Discussion

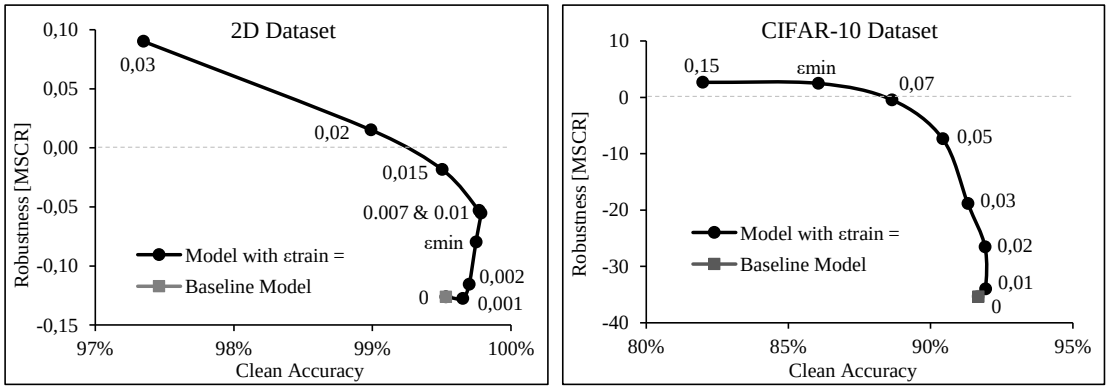
5.1. Applicability of the MSCR metric

Our results from the experiments indicate that the relative difference between the noise-augmented robust accuracy and the clean accuracy is a measure for corruption robustness of models. For $\varepsilon_{test} = \varepsilon_{min}$ in particular, this relative difference that we named MSCR steadily increases



(a) Clean Accuracy (b) Robustness (MSCR)

Figure 5: Model comparison on 2D Dataset with regards to clean accuracy and robustness (MSCR): RF versus 1NN model with different ε_{train}.



(a) 2D Dataset (b) CIFAR-10 Dataset

Figure 6: Accuracy-robustness-tradeoff for models trained with different levels of augmented training noise ε_{train}, compared to the baseline model with $\varepsilon_{train} = 0$. Models with both higher MSCR and higher clean accuracy (when the curve evolves towards the top right corner) contradict the inherent tradeoff.

with higher corruption robustness of the RF model on 2D data and the wide residual network on CIFAR-10. This way, we verify the metric's capability to reflect the corruption robustness of different models. However, this claim is based on the assumption that increasing corruption robustness of our models can be generated through training with higher noise levels. This seems evident based on research by [10], but requires future validation like in [11], who confirm that their Gaussian robustness metric is strongly correlated with the popular physical corruptions benchmark by [8].

On the 2D dataset, the 1NN model shows a constant, superior MSCR value compared to the RF model for all $\varepsilon_{train} \leq 0.007$, where classes are still predominantly separated. This is the performance expected from an inherently robust model such as 1NN, which fits its decision boundary based on maximum class separation. The MSCR values are able to correctly display this interrelation.

5.2. Disadvantages and advantages of the MSCR metric

In our experiments, the steady robustness increase for higher ε_{train} also holds for other levels of testing noise than ε_{min}. The MSCR value, which uses ε_{min}-corruptions as the underlying robustness requirement, is only one particular case of this robustness calculation approach. It has to be emphasized that from our results in Tables 2 and 3, we cannot observe any conspicuities for $\varepsilon_{test} \sim \varepsilon_{min}$.

For example, there is no indication that models perform well below this noise level while massively dropping off at higher noise levels, as could be presumed from the r-separation theory. It is therefore evident to conclude that measuring corruption robustness works with other ε_{test}-values. In practice, if specific corruptions are known for an application, those corruptions should also be used for testing, e.g. through benchmarks [8].

However, we emphasize that the MSCR metric is advantageous in two ways: First, it does not require prior physical knowledge to define corruption distributions, like e.g. [8] does. Instead, it only requires measuring the actual class separation from any classification dataset. Second, the MSCR can be interpreted with a clear contextual meaning, since the robustness requirement is derived from the dataset: It measures "the theoretically avoidable loss (or win) of accuracy due to statistical corruptions".

5.3. On achieving high MSCR values

Clearly, avoiding any loss of accuracy on ε_{min}-noise is hard to achieve in practice on high-dimensional data. For CIFAR-10, $MSCR = 0$ can be achieved, but only with $\varepsilon_{train} = 0.07$, where the clean accuracy declines by 3 percentage points compared to $\varepsilon_{train} = 0$. We also verify our conjecture that $MSCR > 0$ is possible for some robust trained models. For this behavior, we find the discovery in [23] a convincing technical explanation. Misclassified data points tend to lie closer to the decision boundary than correctly classified data points. The data augmentations on a misclassified data point therefore have a high chance of causing a favorable class change. At the same time, data augmentations on correctly classified points have a lower chance of causing an unfavorable class change when their distance to the decision boundary is high, which is what a robust model is trained for.

5.4. The accuracy-robustness-tradeoff

Besides our investigation of the MSCR metric, we report on findings regarding the tradeoff between accuracy and corruption robustness. For both 2D and CIFAR-10 datasets we observe higher clean and robust accuracy on any test noise when training a model with a specific level of uniform noise ($\varepsilon_{train} = 0.007$ for 2D, $\varepsilon_{train} = 0.01$ for CIFAR-10), compared to standard training. For the 2D data, this optimum ε_{train} value is even higher than ε_{min}, the value which the r-separation theory suggests to be beneficial for robustness while not hurting accuracy. This could be due to the major proportion of minimal distances of data points to other classes being significantly bigger than ε_{min}. Our results are statistically significant for the 2D dataset experiment. For 20 runs per trained model on CIFAR-10, we emphasize that claiming higher mean clean accuracy for any $\varepsilon_{train} > 0$ compared to $\varepsilon_{train} = 0$ does not achieve 95%-confidence in a pairwise statistical comparison. More than 20 runs are necessary to obtain statistically significant results, which we could not achieve due to limited computational resources. Hence, we only treat our results on CIFAR-10 regarding the accuracy-robustness-tradeoff as suggestions.

The suggestion that some $\varepsilon_{train} > 0$ leads to higher clean accuracy than $\varepsilon_{train} = 0$ has theoretical relevance. It supports the claim made, but not practically proven by [3], that accuracy and robustness are not in an inherent tradeoff as long as the noise level ε fulfills Equation 1.

The result also seems relevant from a practical perspective, since developers may try some ε_{train} for training data augmentation, which increases robustness without drawbacks regarding accuracy. We emphasize that this practical implication is only valid for the very limited model architectures, datasets and augmentation distributions we tested. For example, our experiments show that noise training below ε_{min} has no effect on an inherently robust model such as 1NN. This is due to the fact that this model type maximizes the class separation of its decision boundary in training anyways.

On the one hand, overcoming the tradeoff for small ε_{train} is not entirely surprising, since it is well known that data transformations and data augmentations can increase generalization of models (in fact, we also used random flips and crops for CIFAR-10 training). [11] and [20] also manage to overcome the tradeoff with more advanced training methods. On the other hand, our results are surprising considering this drawback-free increase in robust accuracy is quite significant for the RF model on 2D data (less than halving the classification error). Also, uniform L_∞ data augmentation is a very simple method and less contextually relevant compared to physically derived augmentations. An explanation may be that the uniform L_p-norm noise allows a stricter coverage of the input parameter space near data points compared to physical data augmentations, enforcing a smooth model that is less prone to overfitting the corruptions.

5.5. Class separation distance for model training

From our results we also need to conclude that in practice, the ε_{min} value has only limited expressiveness when trying to find the optimal ε_{train} with regards to (robust) accuracy. This is visible in Figures 6a and 6b, where based solely on the r-separation theory, we may have expected the curve to reverse its trend along the x-axis when $\varepsilon_{train} = \varepsilon_{min}$. In reality, the best overall accuracy for the 2D data is achieved for $\varepsilon_{train} \sim 2 * \varepsilon_{min}$, while on CIFAR-10 it is achieved for $\varepsilon_{train} < \varepsilon_m in/5$. We suspect that high-dimensional datasets are notoriously hard to train with regards to high robust accuracy, at least for such ε_{min} levels their high L_∞ class separation distance

inevitably entails. We suspect that on other datasets ε_{min} may be even greater and further away from the optimum ε_{train}. Additional research is needed on various distance measures, dataset dimensions and model types in order to utilize class separation distances for optimizing robust accuracy.

5.6. Optima of ε_{train} vs. ε_{test}

Another interesting finding from the accuracy matrix of both datasets is that the best ε_{train} value for models evaluated with certain ε_{test} deviates from the expected diagonal. For example, $\varepsilon_{train} = 0.03$ is not the best choice to prepare for $\varepsilon_{test} = 0.03$. In Figure 7, the accuracy matrix for CIFAR-10 from Table 3 is visualized in a 3D plot, which shows how the optima in (robust) accuracy deviate from the diagonal. It appears that for low noise levels the best choice is $\varepsilon_{train} > \varepsilon_{test}$, while for higher noise levels $\varepsilon_{train} < \varepsilon_{test}$ is more favorable. This suspected dependency needs further investigation.

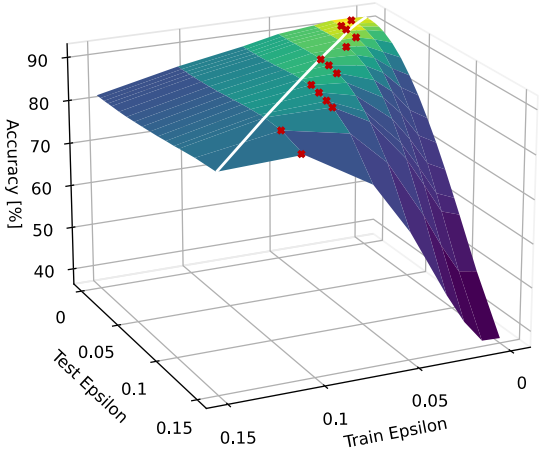


Figure 7: CIFAR-10 (robust) accuracies for different ε_{train} and ε_{test}. The optima, marked with points, deviate from the diagonal (white line where $\varepsilon_{train} = \varepsilon_{test}$): towards higher ε_{train} for lower noise levels and towards lower ε_{train} for higher noise levels.

6. Conclusion

In this article we evaluated a data augmentation method in order to obtain a comparable, interpretable measure of corruption robustness for classifiers. We measured the relative difference between the robust accuracy on corrupted test data and the clean accuracy. We proposed to use half the minimal class separation distance measured from the dataset as the maximum distance ε_{min} of the augmented test noise. This robustness requirement does not presume any prior knowledge about real corruption distances. It theoretically allows a classifier to be fully robust while not losing accuracy. The class separation distance therefore gives our metric a distinct meaning: It represents any "avoidable" loss (or win) in accuracy due to corruptions. We experimentally showed that our metric is able to reflect various degrees of model robustness.

From training classifiers with different levels of noise we found that classifiers with the highest robust accuracy on a certain level of noise are not strictly those, which are trained on this same level of noise. We also presented indications that a tradeoff between accuracy and corruption robustness is not inherent: In our experiments, simple augmentation training on significant random uniform noise could improve test accuracy of classifiers additionally to their robustness, compared with normal training. However, the minimal class separation distance could in practice not guide us towards the optimal values of training noise. These findings regarding the accuracy-robustness-tradeoff could in our opinion be useful in practice.

Our work seems to fit into a gap between those researchers optimizing test accuracy and those optimizing robustness. Our future work will include further investigations of data augmentation training and testing using other dataset types, distance metrics and corruption distributions. It would be of additional interest, whether some increase in adversarial robustness can be obtained without loosing accuracy. Our findings emphasize the potential and encourage the development of advanced training procedures mitigating the accuracy-robustness-tradeoff, since the combination of both properties is essential from a risk assessment perspective.

References

[1] G. Siedel, S. Voß, S. Vock, An overview of the research landscape in the field of safe machine learning, in: Volume 13: Safety Engineering, Risk, and Reliability Analysis; Research Posters, American Society of Mechanical Engineers, 2021. doi:10.1115/IMECE2021-69390.

[2] T.-W. Weng, H. Zhang, P.-Y. Chen, J. Yi, D. Su, Y. Gao, C.-J. Hsieh, L. Daniel, Evaluating the robustness of neural networks: An extreme value theory approach, International Conference on Learning Representations (ICLR) (2018).

[3] Y.-Y. Yang, C. Rashtchian, H. Zhang, R. R. Salakhutdinov, K. Chaudhuri, A closer look at accuracy vs. robustness, Advances in neural information processing systems 33 (2020) 8588–8601.

[4] X. Zhao, W. Huang, V. Bharti, Y. Dong, V. Cox, A. Banks, S. Wang, S. Schewe, X. Huang, Reliability assessment and safety arguments for machine learning components in assuring learning-enabled autonomous systems, arXiv preprint arXiv:2112.00646 (2021).

[5] Deutsches Institut für Normung, Din spec 92001-2: Artificial intelligence – life cycle processes and quality requirements: Part 2: Robustness, 2020.

[6] A. Fawzi, O. Fawzi, P. Frossard, Analysis of classifiers' robustness to adversarial perturbations, Machine learning 107 (2018) 481–508.

[7] J. Gilmer, N. Ford, N. Carlini, E. Cubuk, Adversarial examples are a natural consequence of test error in noise, in: International Conference on Machine Learning, 2019, pp. 2280–2289.

[8] D. Hendrycks, T. Dietterich, Benchmarking neural network robustness to common corruptions and perturbations, International Conference on Learning Representations (ICLR) (2019).

[9] E. Rusak, L. Schott, R. S. Zimmermann, J. Bitterwolf, O. Bringmann, M. Bethge, W. Brendel, A simple way to make neural networks robust against diverse image corruptions, in: European Conference on Computer Vision, 2020, pp. 53–69.

[10] B. Wang, S. Webb, T. Rainforth, Statistically robust neural network classification, in: Uncertainty in Artificial Intelligence (UAI), 2021, pp. 1735–1745.

[11] R. G. Lopes, D. Yin, B. Poole, J. Gilmer, E. D. Cubuk, Improving robustness without sacrificing accuracy with patch gaussian augmentation, arXiv preprint arXiv:1906.02611 (2019).

[12] C. Paterson, H. Wu, J. Grese, R. Calinescu, C. S. Pasareanu, C. Barrett, Deepcert: Verification of contextually relevant robustness for neural network image classifiers, in: International Conference on Computer Safety, Reliability, and Security, 2021, pp. 3–17.

[13] O. Molokovich, A. Morozov, N. Yusupova, K. Janschek, Evaluation of graphic data corruptions impact on artificial intelligence applications, in: IOP Conference Series: Materials Science and Engineering, volume 1069, 2021, p. 012010.

[14] P. Schwerdtner, F. Greßner, N. Kapoor, F. Assion, R. Sass, W. Günther, F. Hüger, P. Schlicht, Risk assessment for machine learning models, NeurIPS 2020 Virtual Workshop: Machine Learning for Autonomous Driving (2020). URL: https://arxiv.org/pdf/2011.04328.pdf.

[15] L. Weng, P.-Y. Chen, L. Nguyen, M. Squillante, A. Boopathy, I. Oseledets, L. Daniel, Proven: Verifying robustness of neural networks with a probabilistic approach, in: International Conference on Machine Learning, 2019, pp. 6727–6736.

[16] Y. Carmon, A. Raghunathan, L. Schmidt, J. C. Duchi, P. S. Liang, Unlabeled data improves adversarial robustness, Advances in neural information processing systems 32 (2019).

[17] J. Cohen, E. Rosenfeld, Z. Kolter, Certified adversarial robustness via randomized smoothing, in: International Conference on Machine Learning, 2019, pp. 1310–1320.

[18] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, International Conference on Learning Representations (ICLR) (2018).

[19] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, M. Jordan, Theoretically principled trade-off between robustness and accuracy, in: International Conference on Machine Learning, 2019, pp. 7472–7482.

[20] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, B. Lakshminarayanan, Augmix: A simple data processing method to improve robustness and uncertainty, International Conference on Learning Representations (ICLR) (2020).

[21] A. Raghunathan, S. M. Xie, F. Yang, J. Duchi, P. Liang, Understanding and mitigating the tradeoff between robustness and accuracy, International Conference on Machine Learning (ICML) (2020).

[22] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, A. Madry, Robustness may be at odds with accuracy, International Conference on Learning Representations (ICLR) (2019).

[23] D. Mickisch, F. Assion, F. Greßner, W. Günther, M. Motta, Understanding the decision boundary of deep neural networks: An empirical study, arXiv preprint arXiv:2002.01810 (2020).