Robustness Margin: A new measure for the robustness of neural networks*

Lionel Kielhöfer[1,2], Annelot W. Bosman[2][0009−0004−1050−5165], Holger H. Hoos[2,3][0000−0003−0629−0099], and Jan N. van Rijn[2][0000−0003−2898−2168]

[1] Intelligence in Quality Sensing, Laboratory for Machine Tools and Production Engineering RWTH Aachen, Campus-Boulevard 30, 52074 Aachen, Germany
lionel.kielhoefer@wzl-iqs.rwth-aachen.de
[2] Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands
{a.w.bosman, j.n.van.rijn}@liacs.leidenuniv.nl
[3] Chair for AI Methodology , RWTH Aachen University, Germany
hh@aim.rwth-aachen.de

Abstract. Neural networks are vulnerable to small input perturbations, which can cause misclassifications to instances that would be correctly classified otherwise. Therefore, assessing the robustness of a neural network is essential in safety-critical applications. Existing robustness measures, such as robust accuracy, fail to capture the robustness of individual inputs to the network and are not easy to interpret, making them unsuitable for comparing different networks. This work introduces a novel robustness measure that addresses these issues by using a probabilistic model of robustness and evaluating its quantiles. Furthermore, we propose both a parametric and a non-parametric estimator to compute confidence bounds for this measure. We evaluate both estimators based on their accuracy and precision over the amount of data used. Both perform reliably given sufficient data; however, the parametric estimator achieves comparable performance with about half as much data, rendering it computationally more efficient.

Keywords: robustness measure · neural network verification · distributions · adversarial robustness

1 Introduction

Knowing whether a neural network is robust and under what conditions is crucial for many use cases, especially as it is known that neural networks can be deceived by slightly altered data during inference. These deviations can occur naturally, as a result of noisy input data, or be induced maliciously when attackers intentionally attempt to cause errors in the network [14,27]. Measuring the

* This research was co-financed by the European Union H2020-MSCA-ITN-2020 under grant agreement no. 956172 (xCTing).

robustness is especially important for safety-critical systems [16], where misclassifications could lead to harmful outcomes. In these situations, a network that is less sensitive to deviations might be preferred over a slightly more accurate one [20].

For image classification, the robustness of a neural network is commonly measured on a per-instance level. Multiple input images are selected, and the robustness of the network against small perturbations on these images is verified. For each image, all deviations within a chosen perturbation size ϵ are considered. If it is not possible to alter the image classification within this ϵ range, the neural network is deemed ϵ-robust on the respective image [5,10,28,4,2], this is also called the local robustness. The largest perturbation size ϵ for which the network maintains ϵ-robustness on the image is called the critical epsilon [2]. Once the ϵ-robustness or critical epsilon is measured across multiple images, various robustness measures can be employed to gauge the overall global robustness of a given network [29,12,14,15,1,22,25,24,6,7,28,27,17]. A commonly used measure is *robust accuracy*, calculated as the percentage of images on which the network demonstrates ϵ-robustness [29,12,14,15,1,22,25,28,27,17]. While approaches that involve altering the model or its training process do exist [9,21], in this work, we consider only robustness measures that evaluate neural networks without requiring any modifications to their architecture or training procedure.

Robust accuracy has several notable drawbacks [2]:

- It is highly dependent on the selected perturbation size ϵ, which requires input from a domain expert to be set to an appropriate value.
- It does not properly capture the local robustness of a network for an individual image, as it requires a predefined perturbation size ϵ and measures the percentage of images that are ϵ-robust.
- These limitations make it difficult to compare the robustness of different networks, as important nuances in their robustness behaviour are not always captured.

Other robustness measures exist that partially address these drawbacks [13,24,6,7,27,28]. However, they still do not enable the effective comparison of robustness between different networks, which is important for neural architecture search [20].

Bosman *et al.* [2] aim to address these drawbacks by introducing robustness distributions. These distributions show how the critical epsilon values are distributed across images for various neural networks. By using the distribution of critical epsilon values, this approach offers a better indication of the robustness of individual images and removes the dependence on a fixed parameter. However, computing these distributions incurs significant computational costs. While they provide valuable statistical insights that can be analysed using standard statistical tools (e.g., the Kolmogorov–Smirnov test [18,26,2]), it is often preferable to summarise robustness with a single representative value to simplify direct network comparisons and interpretation.

In this work, we introduce a robustness measure called the *robustness margin*, which also uses robustness distributions to address several of the limitations of robust accuracy. The robustness margin calculates a predetermined σ-quantile of

the robustness distribution, allowing for a small probability σ that an arbitrary reference image will not be ϵ-robust. It determines the largest perturbation size ϵ consistent with this probability. The parameter σ is task-specific and reflects the desired level of permissible error.

A key advantage of our approach is that it reduces the reliance on a domain expert. Selecting an appropriate ϵ-value typically requires both domain-specific knowledge and expertise in robustness verification. In contrast, the σ-parameter in our method directly corresponds to the acceptable quality threshold, a concept already familiar to experts and routinely used in practice. Since the determination of such thresholds is a well-established task, extensive guidance from existing literature is available, making it straightforward for practitioners to select meaningful σ-values for a wide range of applications.

Additionally, our statistical approach inherently supports the construction of confidence intervals. Rather than relying solely on singular queries to a verification engine, these confidence intervals provide an explicit indication of whether the available data (in terms of samples for which a critical epsilon value has been determined) is sufficient to draw meaningful conclusions about the robustness of the network or if additional verification queries are necessary, saving computational resources. When more samples are being processed, the estimators become more reliable in determining the robustness margin, and the confidence bounds become smaller.

Here, the robustness margin is measured on neural networks trained for image classification tasks. However, the measure and our contributions are not task-specific and are broadly applicable to any task where evaluating the adherence of a given network to an acceptable quality threshold is relevant. Specifically, we make the following contributions:

- **Robustness margin:** We introduce the robustness margin, a novel robustness measure, and provide its formal mathematical definition.
- **Estimation methods:** We propose two distinct estimators for calculating the confidence interval of the robustness margin, providing statistical accuracy in the measure:
  - The *non-parametric estimator* serves as a ground truth of the robustness margin.
  - The *parametric estimator* leverages the findings of Bosman *et al.* [2], which highlight similarities between robustness distributions and log-normal distributions, to increase the computational efficiency.
- **Evaluation of estimators:** We assess both estimators in terms of computational efficiency by examining their accuracy, precision, and effectiveness in selecting the network with the highest robustness margin as the number of samples used for the estimation increases. The samples used for the estimators are the calculated critical epsilon values for different images, which provide insight into the underlying robustness distribution.

We find that the non-parametric estimator exhibits significantly higher variance than the parametric estimator, particularly when fewer than 20 samples are used, at which point the variance becomes too large for it to be an effective

estimator. Beyond this, the parametric estimator is more precise on average, and with around 100 samples, it consistently outperforms the non-parametric estimator in terms of precision.

Additionally, both estimators are accurate across any number of samples, as their confidence intervals tend to overlap with the ground truth. For comparing the robustness margin of two networks, we observe that both estimators are conservative, typically indicating overlap of the confidence intervals rather than incorrectly suggesting that the wrong network is strictly more robust.

In our experiments, the non-parametric estimator requires at least 200 samples to reliably indicate the more robust network, while the parametric estimator achieves this with only 100 samples. Furthermore, when two networks are equally robust, the non-parametric estimator tends to incorrectly suggest a more robust network when fewer than 20 samples are used, highlighting its ineffectiveness at these lower sample numbers. In contrast, the parametric estimator does not suffer from this issue and consistently indicates an overlap, even at lower sample numbers.

2 Background

In this section, we present the formal definitions relevant to the robustness of neural networks. With these definitions, we formalised different robustness measures.

2.1 Local Robustness

One of the most prominent robustness properties is local robustness, which means that a neural network is robust against small input perturbations in a given input. To maintain generality, the norm is not specified in the definitions, as it varies across the literature. Commonly used norms include the l_1, l_2 and l_∞ norms. For the results presented in this paper, the l_∞-norm is used, as most verifiers are compatible with this norm [19]. Table 1 provides a summary of the mathematical notation used in the definitions.

The local robustness property of a neural network for an input can formally be expressed as follows:

Definition 1 (ϵ-robustness). *Consider a neural network $f : \mathbb{R}^n \to \mathbb{N}$ that classifies inputs, a correctly classified reference input $x_0 \in \mathbb{R}^n$, and a perturbation size ϵ. The network is ϵ-robust, or locally robust, on x_0 if $\forall x \in \mathbb{R}^n : ||x - x_0|| < \epsilon \Rightarrow f(x) = f(x_0)$.*

The idea of ϵ-robustness is to analyse all potential perturbations within a distance ϵ from the reference input to determine whether any of them cause the neural network to misclassify. A visual example can be seen in Figure 1. We note that robustness is only considered for originally correctly classified inputs, whilst the robustness of misclassified inputs is not considered in this paper. However,

Table 1. Overview of the most important notation used in this paper

Notation	Meaning
$x \in \mathbb{R}^n$	**Input**. For example, if this were an image, it would consists of n pixels, each taking a value in the set of real numbers. Correctly classified inputs and perturbations are denoted x_0 and x, respectively.
$f : \mathbb{R}^n \to \mathbb{N}$	**Neural Network**. Various types of neural networks exist, but our notation describes neural networks as functions whose output is a natural number indicating the class of the input.
ϵ	**Perturbation size**. Given a correctly classified reference input x_0, the perturbation size ϵ defines the neighbourhood of perturbations x around the reference input that is considered. The set of perturbations is expressed as $\{x : \|x - x_0\| < \epsilon\}$, where $\|\cdot\|$ is a specific norm.

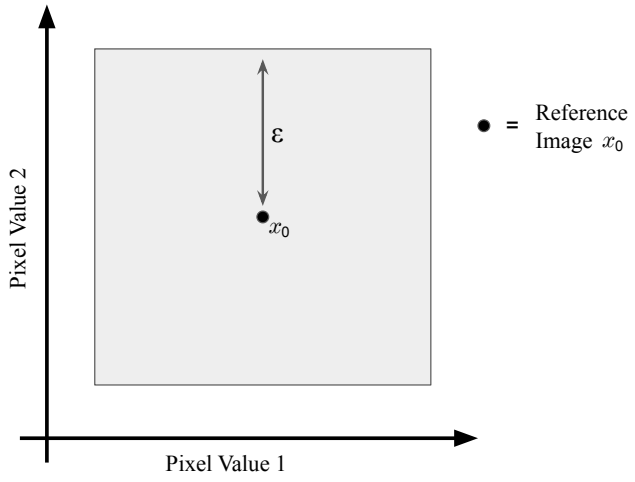


Fig. 1. A 2-dimensional example of verifying whether a correctly classified reference input x_0 is ϵ-robust using the l_∞ norm. For example, if the input were an image, each axis would represent the value of a different pixel, meaning we are examining the space of images composed of only two pixels. The ϵ-square around x_0 is displayed in grey, and x_0 is considered ϵ-robust if each x within the square is assigned the same class as x_0.

we retain the general notation for generality. In our case, $f(x_0)$ is the ground truth label of x_0.

We adopt the critical robustness framework used by Bosman *et al.*[2], according to which the critical robustness property of a neural network for a correctly classified reference input can formally be defined as follows:

Definition 2 (Critical epsilon ϵ^*). *Consider a neural network $f : \mathbb{R}^n \to \mathbb{N}$ that classifies inputs and a correctly classified reference input $x_0 \in \mathbb{R}^n$. The criti-*

cal epsilon, or critical robustness, of the network on that input is the perturbation size ϵ^* such that the network is ϵ-robust on x_0 for all $\epsilon \leq \epsilon^*$, but not ϵ-robust for any $\epsilon > \epsilon^*$.

The critical epsilon represents the maximum perturbation size ϵ for which the network retains ϵ-robustness; it is denoted ϵ^*, or $\epsilon^*_{x_0}$ when the reference input x_0 is not explicitly clear from the context. Note that the critical epsilon ϵ^* is hard to determine exactly; for that reason, we often resort to finding a lower bound on the critical epsilon. Additionally, Bosman *et al.* [2] introduced robustness distributions, which can formally be defined as follows:

Definition 3 (Robustness distribution). *Consider a neural network $f : \mathbb{R}^n \to \mathbb{N}$ that classifies inputs. A robustness distribution represents the probability that the network achieves a critical epsilon ϵ^* for an arbitrary correctly classified reference input x_0. It is typically represented by its cumulative distribution function (CDF):*

$$P(\epsilon^* < \epsilon) := \mathbb{P}(\{x_0 : \epsilon^*_{x_0} < \epsilon\}) \tag{1}$$

where $\epsilon^*_{x_0}$ is the critical epsilon of the reference input x_0. Note that the set of correctly classified reference inputs x_0 is not specified. This is intentional, as it is task-dependent and usually unknown. When necessary, this set will be denoted $\mathcal{X}_0 \subset \mathbb{R}^n$.

2.2 Robust Accuracy

These definitions offer a framework to formally define the commonly used robustness measure: robust accuracy.[4] Reviewing the literature [29,12,14,15,1,22,25,28,27,17], this measure can formally be expressed as follows:

Definition 4 (Robust accuracy). *Consider a neural network $f : \mathbb{R}^n \to \mathbb{N}$ and a perturbation size ϵ. Robust accuracy indicates the probability that the network is ϵ-robust for an arbitrary classified reference input x_0. This will be denoted by:*

$$P(\epsilon) := \mathbb{P}(\{x_0 : ||x - x_0|| < \epsilon \Rightarrow f(x) = y_{x_0}\}) \tag{2}$$

where y_{x_0} is the ground truth label of a reference input x_0. In practice, the following estimator is used to calculate the robust accuracy [29,12,15,1,22,25,28,27,17]:

Estimator 1 (Robust accuracy)**.** Given a set of test data D, which was not used to train the neural network $f : \mathbb{R}^n \to \mathbb{N}$, an estimator for its robust accuracy is obtained by taking the mean over the classified test data D, as follows:

$$\widehat{P(\epsilon)} := \frac{|\{x_0 \in D : \forall x \in \mathbb{R}^n \text{ it holds that } ||x - x_0|| < \epsilon \Rightarrow f(x) = y_{x_0}\}|}{|D|} \tag{3}$$

[4] It is also referred to as the astuteness [29], adversarial error rate [12], adversarial accuracy [28] and certified accuracy [22]. It is also equivalent to 1 minus the adversarial frequency [1].

2.3 Average minimum distortion

A less common robustness measure is the average minimum distortion.[5] Reviewing the literature [24,6,7,27,28,13], this measure can formally be expressed as follows:

Definition 5 (Average minimum distortion). *Consider a neural network $f : \mathbb{R}^n \to \mathbb{N}$. The average minimum distortion represents the expected critical epsilon for an arbitrary correctly classified reference input x_0, denoted as $E(\epsilon^*) := \mathbb{E}(\{\epsilon^*_{x_0} : x_0 \in \mathcal{X}_0\})$, where $\mathcal{X}_0$ is the set of all possible correctly classified reference inputs in the given task.*

To calculate the average minimum distortion, the following estimator is used in practice [13,24,6,7,27,28]:

Estimator 2 (Average minimum distortion)**.** Given a set of correctly classified test data D, which was not used to train the neural network $f : \mathbb{R}^n \to \mathbb{N}$, an estimator for its average minimum distortion is obtained by taking the mean over D, as follows:

$$\widehat{E(\epsilon^*)} := \frac{1}{|D|} \cdot \sum_{x_0 \in D} \epsilon^*_{x_0} \tag{4}$$

The average minimum distortion addresses the first limitation of robust accuracy by not being dependent on a parameter. This eliminates the need for expert domain knowledge. However, the second limitation persists, as it does not provide a reliable indication of the robustness of an individual input. Especially when dealing with robustness distributions with high variance or outliers, the mean can fail to reflect the robustness of individual inputs. Bastani *et al.* [1] address this issue by introducing the concept of adversarial severity. This measure is similar to the average minimum distortion but only takes into account critical epsilons that are below a specific perturbation size ϵ, so $E(\epsilon^* < \epsilon) := \mathbb{E}(\{\epsilon^*_{x_0} < \epsilon : x_0 \in \mathcal{X}_0\})$. However, this reintroduces the need for expert domain knowledge for selecting an appropriate perturbation size ϵ. Just like robust accuracy, these limitations make it challenging to compare the robustness of two networks using the average minimum distortion.

3 Robustness Margin and Estimators

To address the limitations of robust accuracy, we introduce a new robustness measure that is based on robust accuracy and incorporates elements inspired by statistical hypothesis testing. We call this measure the robustness margin. After, we introduce two estimators that determine the robustness margin for a given network.

[5] It is also referred to as the average verified bound [13] and the mean minimum adversarial distortion [28].

3.1 Robustness Margin

The robustness margin allows for a small probability σ that an arbitrary correctly classified reference input will not be ϵ-robust and determines the largest ϵ consistent with this probability. The value of σ should be chosen to be very small. A common choice, often used in statistical hypothesis testing, is $\sigma = 0.05$, which can also be adopted in this context. However, the selection of σ is task-dependent, as some tasks may require a significantly lower permissible error.

Definition 6 (Robustness margin). *Consider a neural network $f : \mathbb{R}^n \to \mathbb{N}$ and a permitted error of $\sigma \in (0, 1)$. The robustness margin indicates the maximum perturbation size ϵ such that the network is ϵ-robust on an arbitrary correctly classified reference input, provided that we allow an error of σ:*

$$\epsilon_\sigma : P(\epsilon_\sigma) = 1 - \sigma \tag{5}$$

Where $P(\epsilon) = \mathbb{P}(\{x_0 : ||x - x_0|| < \epsilon \Rightarrow f(x) = f(x_0)\})$ is the robust accuracy.

The robustness margin returns a value of ϵ for a given permitted error σ, thus removing the dependence on ϵ. With the critical robustness framework from Definitions 2 and 3, this can be reformulated. The following observation can be made about $P(\epsilon)$ from Definition 4:

$$P(\epsilon) = \mathbb{P}(\{x_0 : \epsilon^*_{x_0} \geq \epsilon\}) = 1 - \mathbb{P}(\{x_0 : \epsilon^*_{x_0} < \epsilon\}) = 1 - P(\epsilon^* < \epsilon) \tag{6}$$

This is because a correctly classified reference input that is ϵ-robust is also ϵ-robust for all perturbation sizes smaller than ϵ. Therefore, its critical epsilon is guaranteed to be at least ϵ.

Therefore, the robustness margin can be reformulated as follows:

$$\epsilon_\sigma : P(\epsilon^* < \epsilon_\sigma) = \sigma \tag{7}$$

Therefore, determining the robustness margin corresponds to a quantile estimation problem for the underlying robustness distribution. Figure 2 provides a visual example of how the robustness margin and robust accuracy can be determined from a given robustness distribution.

The robustness margin addresses the disadvantages of robust accuracy as follows:

a. The parameter σ is more intuitive for someone without expertise in safety margins and neural network verification; it is rooted in statistics and directly reflects the tolerance for mistakes. Also known as the acceptable quality threshold, a concept routinely used in practice. This parameter is task-specific and is chosen to reflect the desired level of permissible error. Because it is a widely used concept, extensive task-specific literature already provides guidance on appropriate values of σ.
b. The robustness margin provides a perturbation size ϵ for which an arbitrary input has a probability of $1-\sigma$ to remain robust, offering a reliable indication of the robustness of individual inputs.

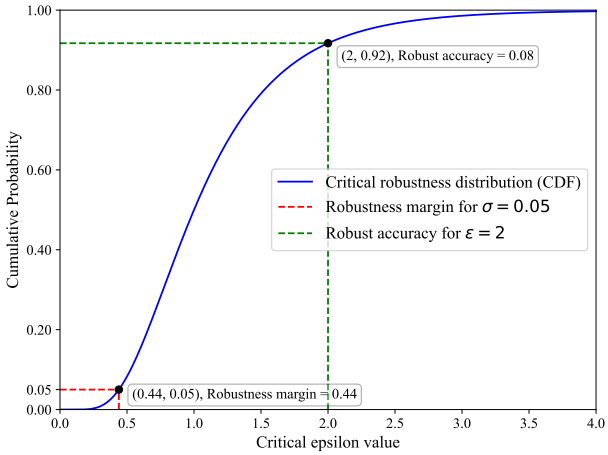


Fig. 2. Illustration of how the robustness margin and robust accuracy can be read of a robustness distribution, whose (hypothetical) CDF is shown in blue. The dashed red lines indicate how the robustness margin for $\sigma = 0.05$ can be read off the plot. The dashed green lines indicate how the robust accuracy for $\epsilon = 2$ can be read from the plot, where the robust accuracy is given by $1 - P(\epsilon^* < \epsilon)$, as shown in Equation 6.

c. By addressing the first two issues, the robustness margin provides a more reliable indicator of a network's robustness behaviour, making it better suited than robust accuracy for comparing the robustness of two networks.

Determining the robustness margin of a neural network for a given σ is equivalent to identifying the σ-quantile of its robustness distribution, as visually represented in Figure 2. In the following, we will present two different methods for empirically determining this value. The first method is a non-parametric approach, which does not make any assumptions about the robustness distribution. The second method is parametric, relying on an assumption about the family of the robustness distribution. The non-parametric approach makes no assumptions about the distribution; however, it requires a significant amount of data to converge to a precise estimate. For the parametric approach to be accurate, a good understanding of the underlying functional family of the distribution is required. For this approach, the robustness distribution is assumed to belong to a functional family, and its parameters are fitted to the data.

This approach requires significantly less data to converge to a precise estimate when compared to the non-parametric approach, provided the distributional assumption holds.

In Section 4, the non-parametric method is used as a baseline to evaluate the reliability of the parametric method. Furthermore, both methods are compared in terms of data efficiency.

3.2 Order Statistics

Order statistics are essential to the non-parametric method and are formally defined as follows:

Definition 7 (Order statistics). *Given a dataset* $\{x_1, \cdots, x_n\}$*, the values are arranged in ascending order to form the ordered set* $\{x_{(1)}, \cdots, x_{(n)}\}$*, where* $x_{(i)}$ *denotes the* i*-th order statistic.*

The following three properties of order statistics, as derived by Meeker *et al.* [23], are relevant:

a. Given a continuous random variable X with CDF P, it follows that $P(X) \sim U(0,1)$, meaning $P(X)$ is uniformly distributed on the interval $[0,1]$.
b. Given that $X_{(i)}$ is the i-th order statistic sampled from $U(0,1)$ with a sample size of n, it follows that $X_{(i)} \sim \text{Beta}(i, n-i+1)$. That is, $X_{(i)}$ has a Beta distribution with parameters i and $n-i+1$.
c. Given the CDF of the Beta distribution with parameters α and $n-\alpha+1$ at a point x, denoted as $\text{Beta}(x; \alpha, n-\alpha+1)$, it holds that $\text{Beta}(x; \alpha, n-\alpha+1) = 1 - \text{Binom}(\alpha-1; n, x)$, where $\text{Binom}(\alpha-1; n, x)$ denotes the CDF of a Binomial distribution with parameters n (number of trials) and x (probability of success in each trial), evaluated at $\alpha - 1$.

Using the first two, the i-th order statistic $X_{(i)}$ sampled from a distribution with CDF P and a sample size of n satisfies

$$P(X_{(i)}) \sim \text{Beta}(i, n-i+1). \tag{8}$$

3.3 Non-parametric Estimator

For the non-parametric estimator, our method follows Meeker *et al.* [23]. Given a dataset of critical epsilons $\{\epsilon_1^*, \epsilon_2^*, \ldots, \epsilon_n^*\}$, the order statistics are given by the ordered set $\{\epsilon_{(1)}^*, \epsilon_{(2)}^*, \ldots, \epsilon_{(n)}^*\}$ and the σ-quantile by $P^{-1}(\sigma)$. A non-parametric confidence interval (CI) for this quantile, with a confidence level of $1-\alpha$, is constructed by selecting appropriate l and u, such that the interval $[\epsilon_{(l)}^*, \epsilon_{(u)}^*]$ has a probability of $1-\alpha$ to contain the σ-quantile. Mathematically, this requires solving the following:

$$\begin{aligned}
\mathbb{P}(\epsilon_{(l)}^* \leq P^{-1}(\sigma) \leq \epsilon_{(u)}^*) &= 1-\alpha, \\
\mathbb{P}(\epsilon_{(l)}^* \leq P^{-1}(\sigma)) - \mathbb{P}(\epsilon_{(u)}^* \leq P^{-1}(\sigma)) &= 1-\alpha, \\
\mathbb{P}(P(\epsilon_{(l)}^*) \leq \sigma) - \mathbb{P}(P(\epsilon_{(u)}^*) \leq \sigma) &= 1-\alpha, \\
\text{Beta}(\sigma; l, n-l+1) - \text{Beta}(\sigma; u, n-u+1) &= 1-\alpha, \\
\text{Binom}(u-1; n, \sigma) - \text{Binom}(l-1; n, \sigma) &= 1-\alpha,
\end{aligned}$$

where the relevant properties related to order statistics have been used. To simplify this even further, only two-sided intervals are considered, meaning that $\mathbb{P}(\epsilon_{(l)}^* \geq P^{-1}(\sigma)) = \mathbb{P}(\epsilon_{(u)}^* \leq P^{-1}(\sigma)) = \alpha/2$. Consequently, the non-parametric estimator is given as follows:

Estimator 3 (Non-parametric confidence interval for the robustness margin)**.** Given a set of test data not used in training the neural network $f : \mathbb{R}^n \to \mathbb{N}$, with corresponding critical epsilons $\{\epsilon_1^*, \epsilon_2^*, \ldots, \epsilon_n^*\}$, we define the non-parametric confidence interval, of at least $1 - \alpha$ level, for the robustness margin at σ as $[\epsilon_{(l)}^*, \epsilon_{(u)}^*]$, where:

$$l = \text{Binom}^{-1}(\alpha/2; n, \sigma) + 1 \tag{9}$$

$$u = \text{Binom}^{-1}(1 - \alpha/2; n, \sigma) + 1 \tag{10}$$

To find a confidence interval for the robustness margin non-parametrically, we order all critical epsilon values from lowest to highest and determine which two indices in this ordered set form a $1-\alpha$ level confidence interval. These indices are given by l and u.

3.4 Parametric Estimator

A key observation by Bosman *et al.* [2] is the similarity between robustness distributions and log-normal distributions. If $\{\epsilon_1^*, \epsilon_2^*, \ldots, \epsilon_n^*\}$ follow a log-normal distribution, then $\{\ln(\epsilon_1^*), \ln(\epsilon_2^*), \ldots, \ln(\epsilon_n^*)\}$ follow a normal distribution. The following two distributions are introduced for the parametric estimator:

a. The standard normal distribution at a point x is denoted $Z(x)$; Z has a mean of 0 and a variance of 1.
b. The noncentral t distribution at a point x is denoted $t(x; n-1, \delta)$, with n-1 degrees of freedom and noncentrality parameter δ.

To estimate the confidence interval for the σ-quantile, with a confidence level of $1 - \alpha$, under the assumption of a log-normal distribution, we use the method described by Meeker *et al.* [23]:

Estimator 4 (Parametric confidence interval for the robustness margin)**.** Given a set of test data not used in training the neural network $f : \mathbb{R}^n \to \mathbb{N}$, with corresponding critical epsilons $\{\epsilon_1^*, \epsilon_2^*, \ldots, \epsilon_n^*\}$ assumed to be log-normally distributed, we define the $1 - \alpha$ level confidence interval for its robustness margin at σ as $[\epsilon_a^*, \epsilon_b^*]$, where:

$$\ln \epsilon_a^* = \mu - t^{-1}(1 - \alpha/2; n - 1, \delta) \cdot \frac{s}{\sqrt{n}} \tag{11}$$

$$\ln \epsilon_b^* = \mu - t^{-1}(\alpha/2; n - 1, \delta) \cdot \frac{s}{\sqrt{n}} \tag{12}$$

with $\mu = \frac{1}{n} \sum_i \epsilon_i^*$, $s = \sqrt{\frac{1}{n} \sum_i (\epsilon_i^* - \mu)^2}$ and $\delta = -\sqrt{n} Z^{-1}(\sigma)$.

Assuming a log-normal distribution of the critical epsilon values, we transform the data into a normal distribution by taking their logarithm. The $1 - \alpha$ level confidence interval of the robustness margin can then be computed using the mean μ, standard deviation s, and non-centrality parameter δ of the resulting normal distribution.

4 Results

The parametric and non-parametric estimators are evaluated in the following three ways, with each evaluation conducted over an increasing number of samples used for the estimator:[6]

- Amount of uncertainty.
- The ability to identify the network with better robustness between the two options.
- The accuracy of the parametric estimator.

4.1 Experimental setup

For the results, we pre-computed[7] critical epsilon values for 1000 instances of 3 networks trained on MNIST [11]. The MNIST dataset consists of handwritten digits along with their corresponding labels, containing 60 000 training instances and 10 000 testing instances. All train instances are used for training the network, and we selected the first 100 test and 100 train instances for each class, leading to 1000 test and 1000 train instances in total, to measure critical epsilons across the selected networks as in [3].

Bosman *et al.* [3] utilise a recent version of the Branch-and-Bound-based neural network verification framework (BaB) [10,5] for verification. For each of the selected instances, for each network, they iteratively make use of k-binary search [2,8] for finding the critical epsilon value. Note that they discretise the search space and investigate a range of ϵ values from 0.001 to 0.4 in intervals of 0.002. Only correctly classified instances for each network were considered for evaluation, potentially resulting in different images selected for each network.

We selected three networks based on the critical epsilon distributions reported by Bosman *et al.* [2]: mnist relu 4 1024, mnist-net 256x4 and mnist-net. mnist relu 4 1024 has the largest average critical epsilon, mnist-net 256x4 has a more typical distribution, and mnist-net has the largest minimal critical epsilon in their study. This way, the three networks form a representative set.

The estimators are evaluated over varying sample sizes, ranging from 10 to 850. For each sample size n, the pre-computed critical epsilon values are used to draw n critical epsilons with replacement. This process is repeated 1000 times, resulting in 1000 different orderings of n instances, and consequently $n \cdot 1000$ critical epsilons. This provides insight into the distribution and performance of the estimators for each sample size, where the sample size represents the number of critical epsilon values available to the estimator.

4.2 Amount of uncertainty

For both estimators, we calculated the mean and quantile information of the estimator to show the uncertainty in the estimation of the robustness margin.

[6] All code is available on: `https://github.com/ADA-research/Robustness-Metric`
[7] For exact computation see: `https://github.com/ADA-research/VERONA`

Naturally, the uncertainty decreases as the critical epsilon of more reference images is determined. Note that these uncertainty calculations are purely intended to display the variability of the estimators over multiple repeated measurements of the robustness margin, while the predictive consequences of these results are discussed in later sections.

In the following, we determine the number of critical epsilon values needed to obtain a certain amount of uncertainty.

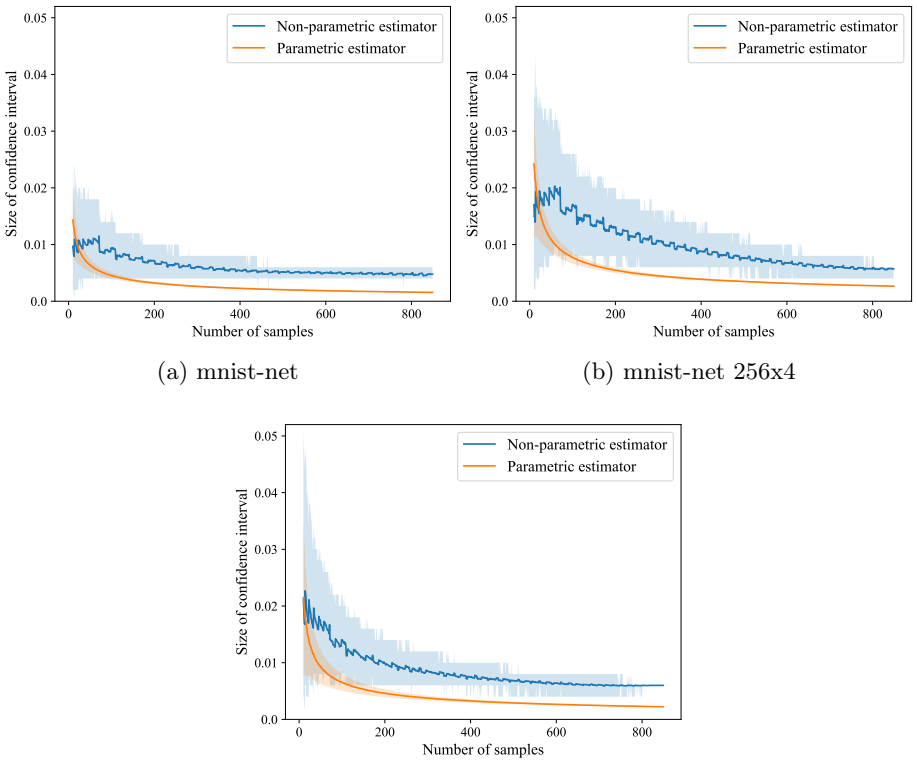
Figure 3 illustrates the mean confidence interval size of the robustness margin over the number of samples, along with the 0.025 and 0.975 quantiles. The samples used for the estimators are the calculated critical epsilon values of different reference images. For each sample size n, 1000 different subsets of size n are drawn from the set of critical epsilons, and their intervals are computed. The difference between the upper and lower bounds of each interval is calculated, and the mean, as well as the 0.025 and 0.975 quantiles of these differences, are determined across the 1000 subsets.

For all three networks, the parametric estimator quickly becomes more precise than the non-parametric estimator as the number of samples increases. With fewer than 20 samples, the non-parametric estimator has a smaller uncertainty on average. However, at these low sample sizes, the quantiles of the non-parametric estimator range from near 0 to 0.04, indicating that the estimator is uncertain at this point. In contrast, the quantiles of the parametric estimator remain much closer together, indicating that it is more certain as it produces a smaller range of values. As the number of samples increases, this trend persists, and by around 100 samples, there is very little overlap between the quantiles of the two estimators. Beyond this point, the parametric estimator can be considered statistically more precise as the amount of uncertainty remains smaller.

Some artefacts can be seen in the plots for the non-parametric estimator; this is due to the discrete nature of the binomial distribution. This discreteness can be seen in Equations 9 and 10, which are used to calculate the indices of the non-parametric interval. To improve smoothness in the curve, a normal approximation of the binomial distribution can be used.

4.3 Identifying the more robust of two given networks

We evaluate the number of samples required for the estimators to reliably determine which of two given networks has a higher robustness margin at $\sigma = 0.05$. Specifically, we assess the chance that each estimator correctly identifies the network with the greater robustness margin as the number of samples increases. As we saw in Figure 3, as the number of samples increases, the estimators become more confident about the robustness margin of the networks. Using a low number of samples, the confidence intervals of the robustness margin for different neural networks will typically still overlap. Note that this is not revealed by this figure, as it merely shows the amount of uncertainty (averaged over all repetitions), and not the predicted mean of the estimators. However, at some point, the uncertainty will be so small that the confidence intervals no longer overlap, and the estimators can determine one of the networks to be most robust. This



(a) mnist-net

(b) mnist-net 256x4

(c) mnist relu 4 1024

Fig. 3. Size of the confidence interval as a function of the number of samples for both estimators. For each sample size, 1000 sets were drawn from the dataset of critical epsilons. The darker line indicates the mean confidence interval size of these sets, while the shaded region represents the range between the 0.025 and 0.975 quantiles for the confidence interval size.

experiment can be repeated various times with various subsets of the data. In the following, we will report for any number of samples the ratio of experiments in which a given estimator selects the most robust network or whether it is not able to select one yet because there is still overlap in the confidence intervals. We expect that, as the number of samples increases, both estimators will be more likely to correctly identify the network with the greater robustness margin. Furthermore, if the parametric estimator is based on an appropriate parametric model, logically, it should reach this conclusion faster than the non-parametric estimator, as it leverages additional information about the distribution.

Figure 4 illustrates the evolution of uncertainty in a single binary comparison, showing both the non-parametric and parametric estimators separately, while Figure 5 summarises the results across all comparisons combining the outcomes

of both estimators in each plot. In this case, minst-net is more robust than mnist-net 256×4, and mnist relu 4 1024 is also more robust than mnist-net 256×4. Additionally, minst-net and mnist relu 4 1024 are equally robust. Note that these results are specific to the chosen σ value. For different values of σ, the relative robustness between networks may change. The optimal outcome, given enough samples, would show no overlap between the confidence intervals and all experiments correctly identifying the more robust network.

The same artefacts mentioned in the previous section can also be seen in these plots. For each sample size n, 1000 different subsets of size n are drawn from the set of critical epsilons, and their intervals are computed. The ratios are calculated based on these intervals.

Both estimators are conservative in their predictions, as they are more likely to indicate overlap rather than incorrectly identifying the more robust network. Figures 5a and 5c illustrate that the parametric estimator correctly identifies the more robust network with fewer samples compared to the non-parametric estimator. The parametric estimator requires approximately 100 samples to, on average, reliably indicate the more robust network, whereas the non-parametric estimator needs at least 200 samples. Additionally, Figure 5b shows that with fewer than 20 samples, the non-parametric estimator struggles to consistently indicate overlap when that is the ground truth. In contrast, the parametric estimator performs consistently in such cases.

From this, a recommended approach for comparing networks is to progressively gather critical epsilon values and recompute the robustness margin until a conclusion can be drawn about which network is more robust, avoiding any unnecessary additional computations. The robustness margin is very unlikely to indicate the wrong, more robust network and will show overlap when more critical epsilon values are needed. Thus, when no overlap is present, the identified network is very likely the more robust one.

4.4 Accuracy of parametric estimator

For both estimators, we calculate the mean and quantile information of the robustness margin to assess their accuracy. An estimator is considered accurate if the calculated quantiles overlap with the ground truth. The ground truth is defined as the robustness margin measured by the non-parametric estimator after 850 samples. We expect the non-parametric estimator to be accurate regardless of the number of samples, as it makes no assumptions about the underlying distribution. Additionally, if the parametric estimator is based on an appropriate parametric model, it should also maintain accuracy for any number of samples.

Figure 6 shows the 0.025 quantile of the lower bound and the 0.975 quantile of the upper bound for the estimated interval of the robustness margin over the number of samples. For each sample size n, 1000 different subsets of size n are drawn from the set of critical epsilons, and their intervals are computed. The 0.025 quantile of the lower bound and the 0.975 quantile of the upper bound for each interval are taken across those 1000 intervals. The result of the non-

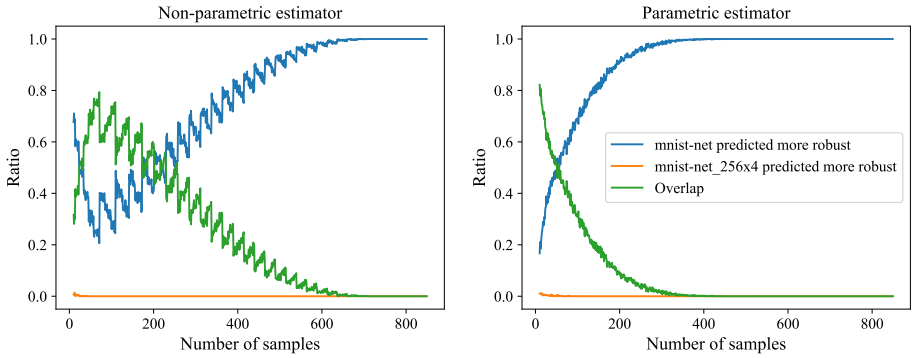


Fig. 4. Evolution of uncertainty for the binary comparison between mnist-net and mnist-net 256x4. The plots show how the proportion of predicted more robust network and overlaps changes as the number of samples increases for the non-parametric estimator (left) and parametric estimator (right).

parametric estimator at 850 samples is used as a ground truth to assess the accuracy of both estimators.

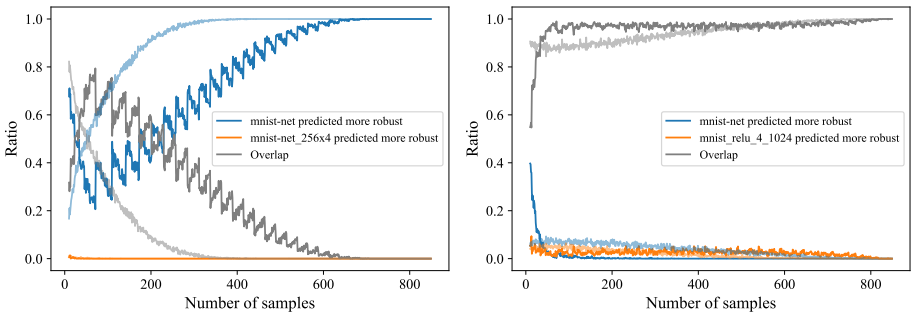
The distribution of intervals for the parametric estimator consistently overlaps with the ground truth, indicating that the parametric estimator provides an accurate estimation of the robustness margin.

5 Conclusion

In this paper, we have introduced a new robustness measure, the robustness margin. Our new measure addresses the limitations of robust accuracy by providing a more intuitive control parameter σ and by offering an interpretable indication of robustness for individual images. It thus provides a solid basis for comparing the robustness of networks. To estimate its confidence bounds in practice, we derived both a parametric and a non-parametric estimator. We evaluated these estimators based on their efficiency in terms of the required number of instances for (i) the ability to identify the more robust of two given networks, (ii) estimator precision, and (iii) estimator accuracy.

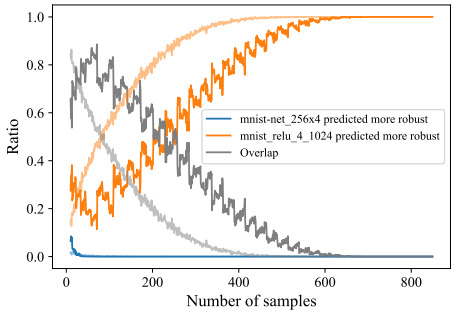
We find that both estimators tend to be accurate for 10 instances. The parametric estimator surpasses the precision of the non-parametric estimator at around 20 instances. However, before reaching this point, the variance of the non-parametric estimator becomes too high for the estimator to be reliable, suggesting that the parametric estimator should be preferred even before 20 instances.

When comparing the robustness of two networks, both estimators are quite conservative, being more likely to indicate a possible tie rather than incorrectly favouring the wrong network. In the scenarios we studied, when one network



(a) mnist-net vs mnist-net 256x4

(b) mnist-net vs mnist relu 4 1024



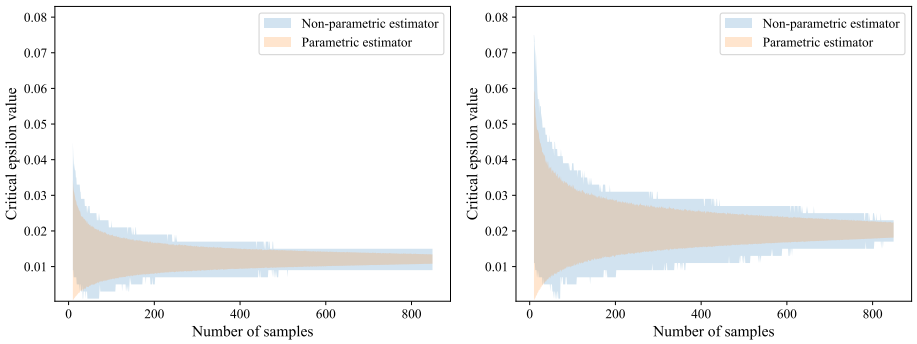
(c) mnist-net 256x4 vs mnist relu 4 1024

Fig. 5. The ratio of network 1 predicted more robust, network 2 predicted more robust, and number of times there is still overlap for each comparison between two networks over a range of sample sizes. We repeated the experiment 1000 times with a different order of instances to account for random effects. Based on these repetitions, we can calculate the ratios. Solid lines represent the non-parametric estimator, and transparent lines represent the parametric estimator.

is more robust, the non-parametric estimator requires at least 200 instances to reliably identify the correct winner, whereas the parametric estimator achieves this with only 100 instances.

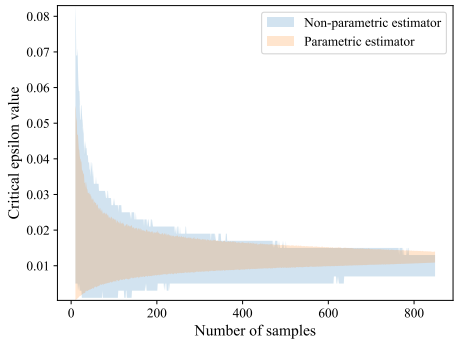
In cases where the networks have a similar robustness margin, the non-parametric estimator has a high chance of incorrectly favouring one network when fewer than 20 instances are used, reinforcing that it is not reliable in this range. The parametric estimator seems to not suffer from this problem with at least 10 instances.

To further validate the reliability and efficiency of the parametric estimator, the same experiments should be conducted on a larger set of networks. Due to computational constraints, evaluations have only been performed on three networks. Expanding the analysis would provide a more comprehensive assessment of its performance across different architectures.



(a) mnist-net

(b) mnist-net 256x4



(c) mnist relu 4 1024

Fig. 6. Distribution of intervals generated by both estimators over several samples. For each sample size, 1000 sets were drawn from the dataset of critical epsilons. The bottom of the shaded region indicates the 0.025 quantile of the lower bounds, while the top indicates the 0.975 quantile of the upper bounds over the 1000 sets.

While in this study, we have only employed complete verification methods, it would be interesting to evaluate our estimators with incomplete verification methods. Using these for estimating critical epsilon values, it may be possible to achieve significant gains in terms of computational efficiency.

References

1. Bastani, O., Ioannou, Y., Lampropoulos, L., Vytiniotis, D., Nori, A., Criminisi, A.: Measuring Neural Net Robustness with Constraints. Advances in Neural Information Processing Systems 29 (NeurIPS 2016) **29**, 2613–2621 (2016)
2. Bosman, A.W., Berger, A., Hoos, H.H., van Rijn, J.N.: Robustness Distributions in Neural Network Verification. Journal of Artificial Intelligence Research (2025), to appear

3. Bosman, A.W., Münz, A.L., Hoos, H.H., van Rijn, J.N.: A Preliminary Study to Examining Per-class Performance Bias via Robustness Distributions. In: International Symposium on AI Verification (SAIV) co-located with the 36th International Conference on Computer Aided Verification (CAV 2024). pp. 116–133. Springer (2024)
4. Botoeva, E., Kouvaros, P., Kronqvist, J., Lomuscio, A., Misener, R.: Efficient Verification of ReLU-based Neural Networks via Dependency Analysis. In: Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI-20). pp. 3291–3299 (2020)
5. Bunel, R., Turkaslan, I., Torr, P., Kohli, P., Mudigonda, P.K.: A Unified View of Piecewise Linear Neural Network Verification. In: Advances in Neural Information Processing Systems 31 (NeurIPS 2018). pp. 1–10 (2018)
6. Carlini, N., Katz, G., Barrett, C., Dill, D.L.: Ground-Truth Adversarial Examples. arXiv preprint arXiv:1709.10207 (2017)
7. Carlini, N., Wagner, D.: Towards Evaluating the Robustness of Neural Networks. In: 2017 IEEE Symposium on Security and Privac (SP). pp. 39–57. Ieee (2017)
8. Cicalese, F., Gargano, L., Vaccaro, U.: On Searching Strategies, Parallel Questions, and Delayed Answers. Discrete applied mathematics **144**(3), 247–262 (2004)
9. Cohen, J., Rosenfeld, E., Kolter, Z.: Certified adversarial robustness via randomized smoothing. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 1310–1320. PMLR (09–15 Jun 2019)
10. De Palma, A., Bunel, R., Desmaison, A., Dvijotham, K., Kohli, P., Torr, P.H., Kumar, M.P.: Improved Branch and Bound for Neural Network Verification via Lagrangian Decomposition. arXiv preprint arXiv:2104.06718 (2021)
11. Deng, L.: The MNIST database of handwritten digit images for machine learning research [best of the web]. IEEE Signal Process. Mag. **29**(6), 141–142 (2012)
12. Dvijotham, K., Stanforth, R., Gowal, S., Mann, T.A., Kohli, P.: A Dual Approach to Scalable Verification of Deep Networks. In: Proceedings of the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2018). pp. 550–559 (2018)
13. Gehr, T., Mirman, M., Drachsler-Cohen, D., Tsankov, P., Chaudhuri, S., Vechev, M.: Safety and Robustness Certification of Neural Networks with Abstract Interpretation . In: 2018 IEEE Symposium on Security and Privac (SP). pp. 3–18. IEEE (2018)
14. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
15. Huang, R., Xu, B., Schuurmans, D., Szepesvari, C.: Learning with a Strong Adversary. arXiv e-prints pp. arXiv–1511 (2015)
16. Julian, K.D., Kochenderfer, M.J., Owen, M.P.: Deep Neural Network Compression for Aircraft Collision Avoidance Systems. Journal of Guidance Control Dynamics **42**(3), 598–608 (2019)
17. Katz, G., Barrett, C., Dill, D.L., Julian, K., Kochenderfer, M.J.: Reluplex: An efficient SMT solver for verifying deep neural networks. In: Proceedings of the 29th International Conference on Computer Aided Verification (CAV 2022). pp. 97–117. Springer (2017)
18. Kolmogorov, A.: Sulla determinazione empirica di una legge didistribuzione. Giorn Dell'inst Ital Degli Att **4**, 89–91 (1933)
19. König, M., Bosman, A.W., Hoos, H.H., van Rijn, J.N.: Critically Assessing the State of the Art in Neural Network Verification. Journal of Machine Learning Research **25**(12), 1–53 (2024)

20. König, M., Hoos, H.H., van Rijn, J.N.: Accelerating adversarially robust model selection for deep neural networks via racing. In: Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI 2024). pp. 21267–21275. AAAI Press (2024)
21. Leino, K., Wang, Z., Fredrikson, M.: Globally-robust neural networks. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 6212–6222. PMLR (18–24 Jul 2021)
22. Li, L., Xie, T., Li, B.: SoK: Certified Robustness for Deep Neural Networks. In: 2023 IEEE Symposium on Security and Privacy (SP 2023). pp. 94–115. IEEE Computer Society (2023)
23. Meeker, W.Q., Hahn, G.J., Escobar, L.A.: Distribution-Free Statistical Intervals, chap. 5, pp. 73–98. John Wiley & Sons, Ltd (2017)
24. Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A.: Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks. In: 2016 IEEE Symposium on Security and Privac (SP). pp. 582–597. IEEE (2016)
25. Shaham, U., Yamada, Y., Negahban, S.: Understanding Adversarial Training: Increasing Local Stability of Neural Nets through Robust Optimization. arXiv preprint arXiv:1511.05432 (2015)
26. Smirnov, N.V.: Approximate Laws of Distribution of Random Variables from Empirical Data. Uspekhi Matematicheskikh Nauk (10), 179–206 (1944)
27. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R.: Intriguing Properties of Neural Networks. In: Proceedings of the 2nd International Conference on Learning Representations (ICLR 2014). pp. 1–10 (2014)
28. Tjeng, V., Xiao, K., Tedrake, R.: Evaluating Robustness of Neural Networks with Mixed Integer Programming. In: Proceedings of the 7th International Conference on Learning Representations (ICLR 2019). pp. 1–21 (2019)
29. Yang, Y.Y., Rashtchian, C., Zhang, H., Salakhutdinov, R.R., Chaudhuri, K.: A Closer Look at Accuracy vs. Robustness. In: Advances in Neural Information Processing Systems 33 (NeurIPS 2020). pp. 8588–8601 (2020)