A Preliminary Study to Examining Per-Class Performance Bias via Robustness Distributions

Annelot W. Bosman[1][0009−0004−1050−5165], Anna L. Münz[2][0009−0007−1328−8698], Holger H. Hoos[1,2,3][0000−0003−0629−0099], and Jan N. van Rijn[1][0000−0003−2898−2168]

[1] Leiden Institute of Advanced Computer Science
Leiden University
The Netherlands
{a.w.bosman, j.n.van.rijn}@liacs.leidenuniv.nl
[2] Chair for AI Methodology
RWTH Aachen University
Germany
{muenz, hh}@aim.rwth-aachen.de
[3] University of British Colombia
Canada

Abstract. As neural networks are increasingly used in sensitive real-world applications, mitigating bias of classifiers is of crucial importance. One often-used approach to controlling quality in classification tasks is to ensure that predictive performance is balanced between different classes; however, it has been shown in previous work that even if class performance is balanced, instances of some classes are easier to perturb in such a way that they are misclassified, which indicates that per-class performance bias exists.
In this preliminary study, we found that even when class performance is balanced, class robustness can vary strongly when assessing the robustness of a given neural network classifier in a more nuanced fashion. For this purpose, we use robustness distributions, i.e., empirical probability distributions of some robustness metric, such as the critical epsilon value, over a set of instances. We observed that the robustness of the same class over the same data can significantly differ from each other for different neural networks; this means that even when a neural network appears to be unbiased, it might be easier to perturb instances of a given class so that they are misclassified.
Furthermore, we explored the robustness distributions when we have a predefined target class, i.e., a specific class into which an instance is misclassified after perturbation. Our empirical results indicate that in most cases, there are significant differences in robustness distributions for different classes.
While our empirical results reported here are for MNIST classifiers, we are currently performing experiments using the German Traffic Sign Recognition Benchmark. Furthermore, we are running experiments with retrained networks for fairness, to see whether this has a significant effect on the per-class robustness distributions. Lastly, we aim to create a robust class fairness metric based on our findings.

Keywords: neural network verification · adversarial robustness · fairness · distribution analysis

1 Introduction

Machine learning research is becoming increasingly fairness-aware. A prominent example of a desirable property of a classifier is an equal performance of all subgroups based on some protected property [1], as an imbalance can pose a threat to fairness. Moreover, predictions obtained from neural networks are known to be vulnerable to slight alterations to otherwise correctly classified inputs, leading to incorrect predictions [8]. Benz *et al.* [1] have shown that the robustness of a neural network against these small alterations can differ across classes, even if the dataset the network was trained on has an equal number of instances for each class. This imbalance in robustness of different classes can have severe practical implications.

Formal verification techniques (such as α, β-CROWN [22]) can be employed to rigorously assess the robustness of neural networks against such input perturbations. Given a neural network and an observation from the dataset, these techniques determine whether a perturbation of at most magnitude ε exists that can force the network to misclassify the given observation. Therefore, the robustness of a network can be expressed in terms of *robust accuracy*, based on how many of the observations will be correctly classified, regardless of adversarial input perturbations within a given radius ε.

In previous work, we introduced robustness distributions [2]. Rather than the commonly addressed yes-or-no question of whether a perturbation of at most ε can lead to a misclassification of a given input, robustness distributions indicate for each input the exact magnitude of perturbation, indicated by ε^*, to which the classification remains robustly accurate. This gives a more balanced view of the robustness of a network, and, unlike robust accuracy, it is not determined based on a single fixed ε.

Recent work has demonstrated that different classes can have different predictive performance and robustness performance [1, 10]. These studies evaluated whether an instance is robust with respect to a fixed pre-defined maximum perturbation level ε using non-formal methods, such as adversarial attacks [11, 13].

This per-class robustness is highly important, for the following reasons:

- **Fairness:** It would be problematic if an entity-recognition system would be less precise at predicting certain classes compared to others. Therefore, imbalanced class performance can be a problem for the fairness of a classifier.
- **Misclassification cost:** Some types of misclassification have more severe consequences than others. One can imagine it is far more problematic when a traffic sign indicating a 130 km/h speed limit gets misclassified as a stop sign than it is when it gets misclassified as a 120 km/h speed limit.

For these reasons, we investigate the robustness distributions at the class level. Confirming significant discrepancies in robustness on a per-class level would have consequences on network selection for practical applications. A practitioner that would have to choose between several options can utilise our approach to make an informed decision based on these per-class distributions regarding the question of which of several networks would be most adequate for a given application. This distinction per class can be considered for both the correct class of an instance, as well as for the

class a given instance is misclassified as; we will refer to the latter concept as the *target class*. We distinguish two specific types of verification:

- *one-to-any verification* determines the robustness of an instance, regardless of which class this given instance gets misclassified as. This concept has been widely studied in the literature (see, e.g. [3, 9]).
- *one-to-one verification* determines for a given (original) class and a given target class up to what perturbation magnitude a given instance cannot be misclassified as this specific target class. Note that it could be possible that with a smaller perturbation radius, the instance could be non-robust against being misclassified as other classes.

By performing these types of verification for a set of instances, grouped per class, we can determine robustness distributions on a per-class level. These distributions consist of the individual approximate perturbation magnitude $\tilde{\varepsilon}^*$ an instance can withstand without being misclassified. While both concepts have well-known equivalents, to the best of our knowledge there has only been limited research towards per-class analysis in the context of robustness. We note that the per-class robustness distributions resulting from the one-to-any verification are related to the per-class recall, whereas the per-class robustness distributions resulting from the one-to-one verification are related to the confusion matrix.

In this study, we make the following contributions:

- We introduce the notion of per-class robustness distributions, using the previously mentioned concepts of one-to-any verification and one-to-one verification.
- We analyse the per-class robustness for three MNIST networks using one-to-any verification. We establish that for most of these networks, the robustness distributions differ significantly across classes.
- We establish that across networks, the robustness of a given class can differ significantly.
- We analyse for a given class the robustness towards all individual target classes for three MNIST networks. We establish that the robustness distributions differ significantly for most of these target classes.
- We establish that across networks, the robustness of a given class towards another target class can differ significantly.

We believe that our preliminary work presented here provides a good basis for connecting robustness distributions to fairness notions and creating a robust class fairness metric.

The remainder of this report is structured as follows. We discuss the main concepts relating to neural network verification, fairness and robustness in the context of classification tasks in Section 2. In Section 3, we discuss the setup of our experiments. Following this, Section 4 presents and discusses our empirical results. Finally, in Section 5, we briefly summarise our findings and discuss the next steps we are currently pursuing, building on our work presented here.

2 Background

In the following, we review aspects of local robustness verification and robustness distributions that form the basis for our work. We also briefly discuss bias and fairness in machine learning.

Neural network verification. Verification methods for neural networks often focus on local robustness properties for image classifiers. When a classifier is locally robust for an input and a predetermined perturbation radius ε, it means that there exists no perturbation of magnitude up to ε that is applied to the input, which will lead to misclassification. The feasible region of perturbations is determined by some norm, in our case the l_∞-norm.

This property can be modelled using mixed-integer linear programming problems [19]; however, due to the nonlinearity of neural networks, these problems can be computationally expensive to solve.

Robustness distributions. In this work, we make use of the concept of robustness distributions introduced by Bosman *et al.* [2]. Robustness distributions are the empirical distributions of the critical ε values, *i.e.*, the largest perturbation radius, over a given set of inputs such that each input is still classified provably accurately. In this work, we use the approximate critical ε, expressed by $\tilde{\varepsilon}^*$, which is found by performing repeated complete verification using k-binary search [2,5,6] for each individual input.

Robustness distributions are used to obtain a nuanced view of the robustness of a given neural network. Instead of computing the robust accuracy of a network for one predetermined ε, the distributions reflect the robustness for an entire range of ε values. Based on the robustness distributions of a network, we can determine the robust accuracy at any given perturbation radius ϵ.

Robust accuracy and robust recall. In many works considering robustness, the robustness of a network is defined by means of *robust accuracy* [7, 16, 21]. Robust accuracy is defined as the percentage of inputs that will be classified correctly, regardless of what perturbation within the predefined bound ε will be applied to any given input. In some cases, including our work presented here, this definition is adapted to ignore originally misclassified inputs [20].

Accuracy is defined over the entire set of inputs, while here, we consider input-specific classes. This means we should consider *recall* rather than accuracy. For this reason, we introduce the concept of *robust recall* on a given class, which we define as the percentage of inputs of that given class, that will be classified correctly regardless of adversarial perturbation applied to this input up to the predefined bound ε.

Bias and fairness. There is a considerable amount of literature on identifying, categorising and mitigating bias in machine learning (for a comprehensive overview, see, *e.g.*, [12, 18]). There is no universal definition of bias, nor a clear distinction between bias and fairness in some of the literature. We refer to bias as a preference of the model that is not solely based on the observed data [14]. Hence, bias describes a possible state of a lack of fairness and primarily only identifies the condition that needs to

be improved. Several metrics have been introduced to identify and quantify a lack of fairness (see, *e.g.,* [4, 12]). In the context of real-world data, such metrics are defined based on protected characteristics of individuals that may lead to discrimination.

Benz *et al.* [1] found that there exist differences between the predictive performance and robustness of different classes. The bias researched in this context is the disparity of predictive performance and robustness based on a fixed maximum perturbation level ε on a per-class level. We extend the work of Benz *et al.* [1] by providing a more detailed analysis using robustness distributions and comparing across different networks. Rather than measuring the per-class robustness at a fixed perturbation magnitude ϵ, these distributions can determine the per-class robustness at any given perturbation magnitude ϵ.

3 Setup of experiments

Network selection. We investigate three conventionally pre-trained MNIST classification networks, with ReLU activation functions. We have selected these networks based on the work by Bosman *et al.* [2], namely the network with the highest median $\tilde{\varepsilon}^*$ on training data, the network with highest minimum $\tilde{\varepsilon}^*$ on training data and the network with highest minimum $\tilde{\varepsilon}^*$ on testing data. Selecting a subset of the networks considered by Bosman *et al.* was necessary because of the computational demands of our analysis. The per-class performance and the number of training and testing instances used for each network are described in Appendix A, Table 1.

Input selection. In this work, we investigate the robustness distributions for the different classes a network is trained for. We randomly selected 100 images for each of 10 original MNIST classes for both testing and training data and thus obtained 1 000 images from the training data and 1 000 images from the testing data. In the following, we refer to an instance as the combination of a network and an image. We removed the images that were misclassified by a given network from the robustness distribution, as they have essentially a critical epsilon value $\tilde{\varepsilon}^*$ of 0. This implies that the set of images in the distribution might vary over networks, as can be seen in Appendix A, Table 1. Of course, when calculating the robust accuracy or robust recall based on the robustness distributions, these images should be considered as well.

We also investigate whether the robustness of a certain class against perturbations to specific other classes can significantly differ for a network. We chose to take 100 instances each from testing and training images with the original label 9. The decision to investigate label 9 was based on the work of Zhang and Evans [23], who investigated the robust error of different classes; from their results, we concluded that the robust test error for original class 9 and with individual other MNIST target classes have the most diverse error rate over the various target classes. For conciseness and computational complexity reasons, we have restricted this part of our analysis to a single original class.

Algorithm setup. To obtain the robustness distributions, we calculate a strict lower-bound on the critical ε value, denoted by $\tilde{\varepsilon}^*$, for each input image, following Bosman *et*

al. [2]. We discretise the problem into intervals of 0.002 in a range of $[0.001, \ldots, 0.800)$ and find $\tilde{\varepsilon}^*$ by performing k-binary search with $k = 2$, which entails that we verify k queries with different ε values for the same input image at the same time [2]. We determine $\tilde{\varepsilon}^*$ by finding two adjacent perturbation radii a, b such that $a < b$, where a verification query with perturbation radius a results in proof that the network is robust and with perturbation radius b an adversarial example is found. In theory, we should obtain $b = a + 0.002$; however, in practice, some verification queries lead to time-outs or out-of-memory errors (in our experiments, this happened in approximately 2% of the instances).

We elected to use k-binary search (with $k = 2$) based on the results of Bosman *et al.* [2]. The intuition behind this is as follows. Consider the situation in which 2 verification queries are running simultaneously for perturbation radii a, b on the same instance such that $a < b$. Then, k-binary has the advantage that if we find that for perturbation radius a the network is non-robust for a certain input, the network is indeed also non-robust for b; we can thus terminate the verification query with radius b, which potentially saves running time. The opposite also holds; if we find that the network is robust against perturbation radius b for a certain input image, it must also be robust for perturbation radius a.

For both one-to-any and one-to-one verification, we use the verification method oval-bab [15] with a time-out of 3 600 seconds per verification query. We selected oval-bab as it is a state-of-the-art verifier and it is relatively simple to adapt the verification property to one-to-one verification within the oval-bab framework.

Execution environment. All our experiments were carried out on a cluster of machines, each equipped with 2 Intel Xeon E5-2683 CPUs with 32 cores, 40MB cache size and 94GB of RAM, running CentOS Linux 7. The amount of RAM available for each verification query for one image and network was set to 20GB. we always used one dedicated CPU core per verification query and restricted each verification query to a time budget of one CPU hour.

4 Results

We now report the results from our empirical analyses. First, as a basis for comparison with the per-class robustness distributions at the core of this work, in Section 4.1, we show general robustness distributions over all instances for the networks included in this work. Then, in Section 4.2, we take a more detailed view and analyse the per-class robustness distributions, in order to determine whether per-class discrepancies in adversarial robustness exist. Next, we investigate whether there are significant differences in the robustness distributions for the same data and class, but different neural networks. The existence of such differences would indicate that, when assessing the robustness of a neural network classifier, one should not merely focus on overall robustness, but also determine whether the most important classes are sufficiently robust against small input perturbations.

Furthermore, in Section 4.3 we considered the robustness of instances from a given class against misclassification into a specific target class (*i.e.,* one-to-one verification),

to investigate whether there is a significant difference in robustness distribution for different target classes and to compare the robustness of target classes for different networks. This can be important in practice when misclassification due to small input perturbations can have more severe effects for some target classes than for others.

4.1 Robustness distributions

In this section, as an introduction to the concept, we report the robustness distribution for each network that is investigated in this work, similar to how they are presented by Bosman *et al.* [2]. Robustness distributions in this context consist of the $\tilde{\varepsilon}^*$ over a set of inputs, where the original class is not of relevance and no target class is specified. This type of robustness distributions give a general, but nuanced, overview of the robustness of a network.

Figure 1 shows the robustness distributions for the aggregated $\tilde{\varepsilon}^*$ over all images from any class for one-to-any classification. Each robustness distribution consists of the $\tilde{\varepsilon}^*$ for 1 000 instances, minus the number of misclassified instances (see Appendix A, Table 1). Naturally, these misclassified instances have a $\tilde{\varepsilon}^*$ of 0.

From the robustness distribution of a given network and the number of misclassified inputs, the robust accuracy for any ε can be obtained as the fraction of correctly classified instances at ε. For example, at $\varepsilon = 0.012$ *mnist_net* and *mnist_relu_4_1024* have a robust accuracy of 94.8 and 93.9 percent, respectively, while at $\varepsilon = 0.04$, they have a robust accuracy of 40.4 and 71.9, respectively.

4.2 Per-class one-to-any verification

In this section, we focus on sets of inputs from one specific target class at a time, while previously, we considered inputs from all classes. Our goal in the analysis conducted here was to determine whether the robustness distributions for different classes differ from each other, and furthermore to investigate how the robustness distributions for different networks for the same class relate to each other.

Figures 2a, 2b and 2c each show the robustness distributions for three different neural networks. In each figure, we show the boxplots for each original MNIST class (digits 0 to 9) from training and testing data, respectively, for a specific neural network. In many cases, the training data distribution looks very similar to the testing data distribution, and when performing a Kolmogorov-Smirnov test with $\alpha = 0.05$, in a large majority of the cases no significant difference between the testing and training distributions was detected (see Appendix B.2, Figure 9).

Specifically, for *mnist_relu_4_1024*, *mnist-net_256x4* and *mnist-net* of the 10 classes investigated overall, 2, 4 and 1, respectively, of the classes showed significantly different robustness distributions for testing and training data.

The distributions for different original classes for the same network were significantly different for both training and testing data, as determined by the Kolmogorov-Smirnov test. This confirms that in addition to the fact that robust recall varies greatly over classes, the robustness distributions do as well.

Figures 3a and 3b show the cumulative distribution function (CDF) plots for the three MNIST networks for original classes 2 and 6, respectively. We found evidence

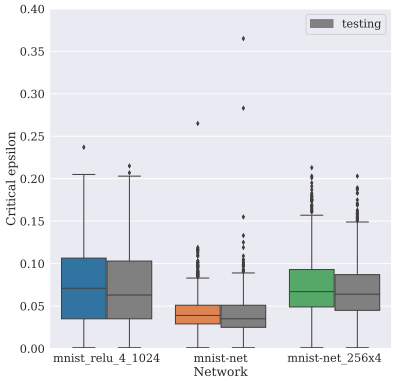


Fig. 1: Boxplot of the distributions of $\tilde{\varepsilon}^*$ for 3 MNIST classifiers. For each, the robustness distribution is shown for correctly classified inputs on training and testing data. We note that the individual distributions do not include images that were originally misclassified by the network, and that each of these distributions is comprised of robustness information of distinct images. The robustness distribution of the networks, for both testing and training data, are significantly different from each other, according to the Kolmogorov-Smirnov test at a standard significance level of $\alpha = 0.05$ (see Appendix B.1, Figure 8). For *mnist-net*, the robustness distributions for training and testing data are significantly different from each other, while for the other two networks, we did not detect statistically significant differences.

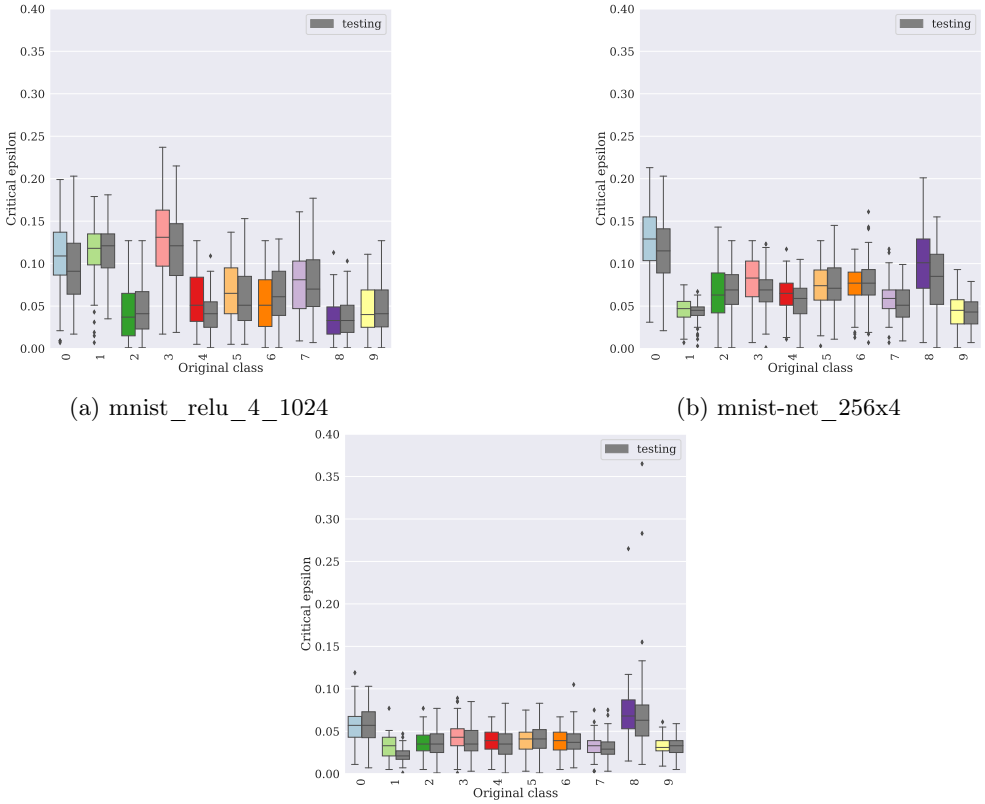
that the robustness distributions for these networks, for both original classes, on testing and training data, are all significantly different, according to a Kolmogorov-Smirnov test with $\alpha = 0.05$ (see Appendix B.3, Figure 10).

Figures 4a and 4b show an instance-level comparison of $\tilde{\varepsilon}^*$ for network *mnist_relu_4_1024* and *mnist-net* for original classes 2 and 6, respectively. Especially for class 6, for which the CDF is shown in 4b, the instances that are easier to perturb for one network are not necessarily the ones that are easy to perturb for the other.

4.3 Per-class one-to-one verification

We now report the results from our experiments on one-to-one verification, where we consider misclassifications to specific target classes. Our goal was to investigate whether the robustness distributions for one-to-one verification differ from each other, as well as assess the differences in one-to-one verification for the same target class between different networks.

Figures 5a, 5b and 5c show the one-to-one robustness distributions for three different networks. Each figure shows the boxplots for a specific MNIST target class, except class 9, which was the original class of the images, in the dataset for training and testing data, respectively. In many of the cases, the training data distribution looks very similar to the testing data distribution, and when performing a Kolmogorov-Smirnov



(a) mnist_relu_4_1024

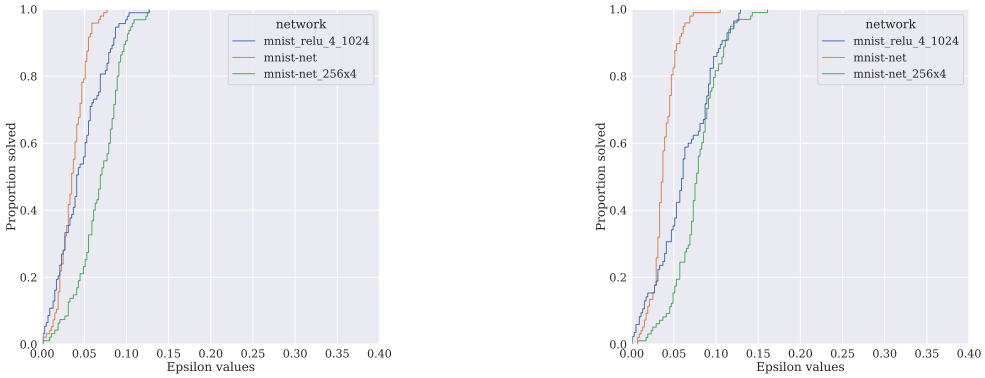
(b) mnist-net_256x4

(c) mnist-net

Fig. 2: Boxplots of the distributions of $\tilde{\varepsilon}^*$ for 3 MNIST classifiers. For each, the robustness distribution is shown for each original class over the respective sets of correctly classified inputs on training and testing data. We note that the individual distributions do not include images that were originally misclassified by the network, and that each of these distributions is comprised of robustness information of distinct images.

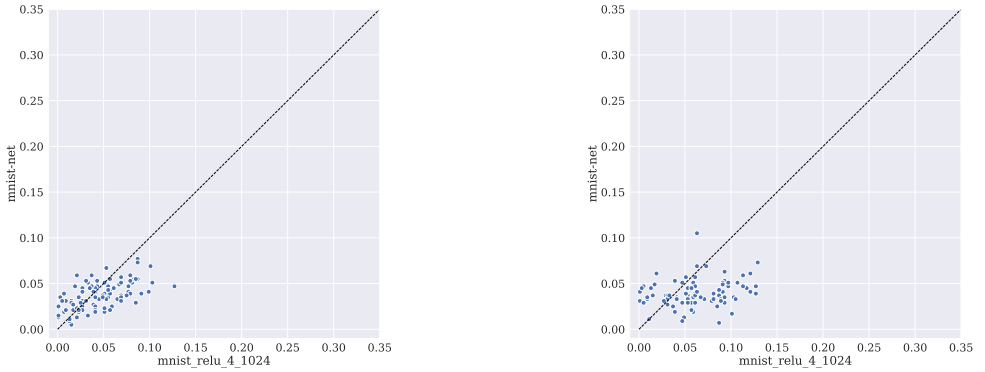
test with $\alpha = 0.05$, in a large majority of the cases, no statistically significant difference between the testing and training distributions was found. Specifically, for *mnist-relu_4_1024*, only the training and testing distribution for target class 0 are significantly different, for *mnist-net_2564*, there was a significant difference for both target classes 3 and 8, and for *mnist-net* there was also a significant difference for target class 3 (see Appendix B.4, Figure 11).

The empirical data suggests that the robustness against targeted perturbations (*e.g.*, adversarial attacks) for different targets with the same original class can be significantly different. In many cases, the robustness distributions for different target classes are significantly different from each other, as determined by the Kolmogorov-



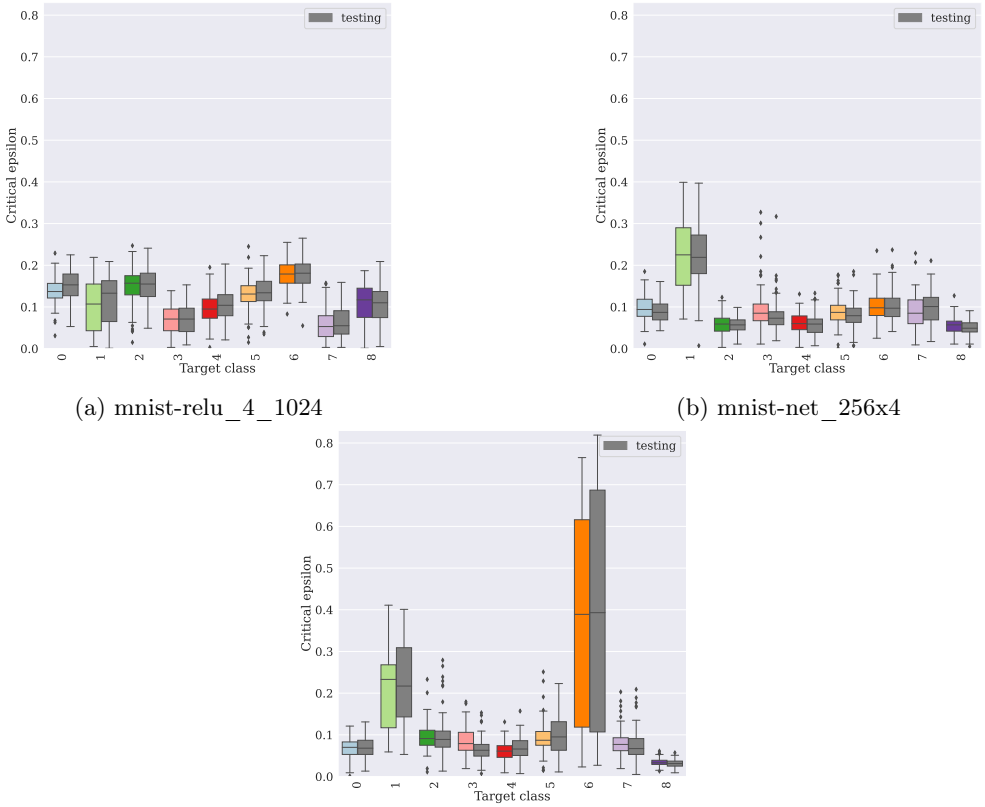
(a) Original class 2 (b) Original class 6

Fig. 3: Empirical CDF plots for the empirical robustness distributions for the 3 investigated MNIST classifiers for 2 different original classes with both quite average behaviour on testing data.



(a) Original class 2 (b) Original class 6

Fig. 4: Scatterplot of $\tilde{\varepsilon}^*$ for different testing images for two original classes for different networks. Each point corresponds to one image. The images that were originally misclassified by at least one of the networks are not included.



(a) mnist-relu_4_1024

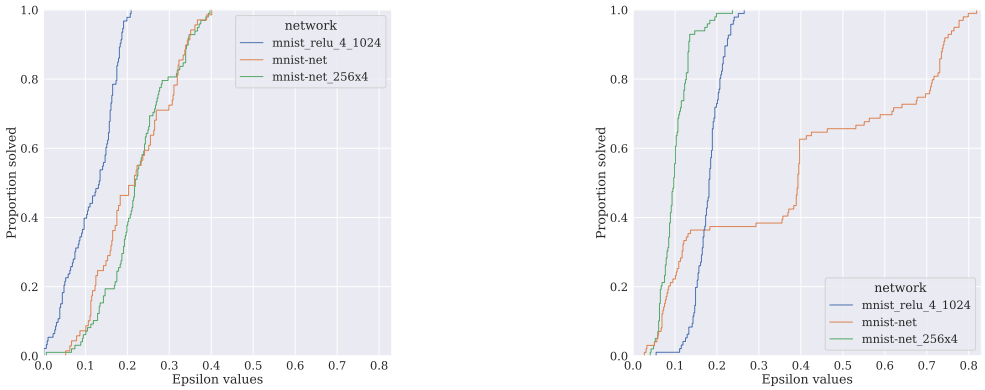
(b) mnist-net_256x4

(c) mnist-net

Fig. 5: Boxplots of the distributions of $\tilde{\varepsilon}^*$, for 3 MNIST classifiers. In each case, the robustness distribution of original class 9 with all different target classes is shown on correctly classified inputs from training and testing data, respectively. We note that the individual distributions do not include images that were originally misclassified by the network and that each of these distributions is comprised of robustness information on the same images.

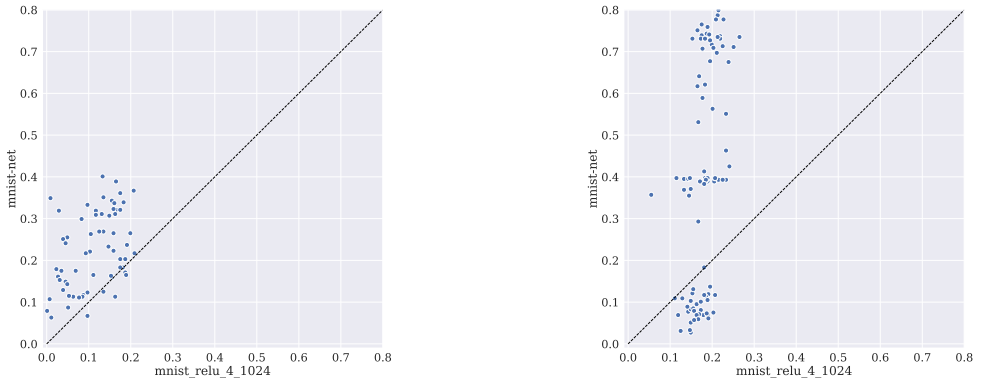
Smirnov test (see Appendix B.5, Figure 12), Figures 6a and 6b show the CDF plots for the three MNIST networks for target classes 2 and 6, respectively.

Figures 7a and 7b show an instance-level comparison of $\tilde{\varepsilon}^*$ for network *mnist_relu_4_1024* and *mnist-net* for target classes 1 and 6, respectively. Especially for target class 6, for which the instance-level comparison is shown in 7b, the $\tilde{\varepsilon}^*$ for *mnist-net* has a much wider range than for *mnist_relu_4_1024*. There also appear to be three clusters, which can also be noticed in Figure 6b, in the form of multiple modes in the CDF plots. Further analysis needs to be performed to determine a possible cause for this.



(a) Target class 1 (b) Target class 6

Fig. 6: Empirical CDF plots for the empirical robustness distributions for the 3 investigated MNIST classifiers for 2 different target classes with typical (left) and atypical behaviour (right) on testing data. For both target classes, the robustness distributions for all different network combinations are from different distributions according to the Kolmogorov-Smirnov test with $\alpha = 0.05$ for both testing and training data.



(a) Target class 1 (b) Target class 6

Fig. 7: Scatterplot of $\tilde{\varepsilon}^*$ for different testing images for two target classes for different networks. Each point corresponds to one image. The images that were originally misclassified by at least one of the networks are not included.

5 Conclusions and Future Work

In this preliminary work, we investigated robustness distributions on the class level. Specifically, we created robustness distributions per class for one-to-any verification, as well as distributions for a given (original) class to a given target class, for one-to-one verification. In particular, for different original classes as well as for different target classes we found significant differences in the robustness distributions, indicating significant per-class bias in the robustness of these networks. To the best of our knowledge, robustness distributions have not been used to study the robustness of networks on a per-class level; we find them useful because they provide additional information to practitioners in need of assessing neural networks based on per-class robustness properties. Additionally, we believe to be the first to study the robustness of neural networks for different target classes.

We show, for the first time, that for the same data from the same original classes, the robustness distributions can also differ significantly between networks: One network can be more robust for inputs from a certain original class than another network, and simultaneously be less robust for another original class. This implies that practitioners should not only consider the overall robustness of networks, but also the robustness of individual classes, especially when dealing with cases where the misclassification cost differs between classes.

Similarly, we found that for the same data, and the same target classes, there can be significant differences in the robustness for different networks. This is particularly interesting when misclassification to certain target classes has more severe practical effects than misclassifications to other target classes.

Currently, we are extending our analysis beyond MNIST and specifically studying the German Traffic Sign Recognition benchmark [17]. Even though our study on MNIST presented here has clearly illustrated the usefulness of robustness distributions for assessing the robustness of different classes and networks, we further plan to extend our work to real-world problems where misclassification costs are non-symmetrical and therefore need to be considered carefully. In addition, we are planning to investigate whether methods that retrain networks for balanced per-class robustness can alter the robustness distributions and potentially eliminate or at least reduce per-class bias neural network robustness.

Lastly, we plan to extend our work to a face recognition benchmark and to analyse the robustness distributions of different classes when sensitive attributes are of concern. We ultimately aim to introduce fairness metrics that also take into consideration robustness. As state-of-the-art classifiers used in this field are potentially too complex to be handled by current complete verifiers, incomplete verification techniques might have to be used in this context.

Acknowledgements

This research was partially supported by TAILOR, a project funded by EU Horizon 2020 research and innovation program under GA No. 952215.

References

1. Benz, P., Zhang, C., Karjauv, A., Kweon, I.S.: Robustness May Be at Odds with Fairness: An Empirical Study on Class-wise Accuracy. In: NeurIPS 2020 Workshop on Preregistration in Machine Learning. pp. 325–342. PMLR (2021)
2. Bosman, A.W., Hoos, H.H., van Rijn, J.N.: A Preliminary Study of Critical Robustness Distributions in Neural Network Verification. In: 6th Workshop on Formal Methods for ML-Enabled Autonomous Systems (FoMLAS) (2023)
3. Brix, C., Bak, S., Liu, C., Johnson, T.T.: The Fourth International Verification of Neural Networks Competition (VNN-COMP 2023): Summary and Results. arXiv preprint arXiv:2312.16760 (2023)
4. Caton, S., Haas, C.: Fairness in Machine Learning: A Survey. ACM Computing Surveys **56**(7), 1–38 (2024)
5. Cicalese, F., Gargano, L., Vaccaro, U.: On Searching Strategies, Parallel Questions, and Delayed Answers. Discrete Applied Mathematics **144**(3), 247–262 (2004)
6. Cicalese, F., Vaccaro, U.: Binary Search with Delayed and Missing Answers. Information Processing Letters **85**(5), 239–247 (2003)
7. Cullina, D., Bhagoji, A.N., Mittal, P.: Pac-learning in the presence of evasion adversaries. In: Advances in Neural Information Processing Systems 31 (NeurIPS 2018) (2018)
8. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and Harnessing Adversarial Examples. arXiv preprint arXiv:1412.6572 (2014)
9. 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)
10. Ma, X., Wang, Z., Liu, W.: On the Tradeoff Between Robustness and Fairness. Advances in Neural Information Processing Systems 35 (NeurIPS 2022) pp. 26230–26241 (2022)
11. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards Deep Learning Models Resistant to Adversarial Attacks. arXiv preprint arXiv:1706.06083 (2017)
12. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: A Survey on Bias and Fairness in Machine Learning. ACM Computing Surveys **54**(6), 1–35 (2021)
13. Mirman, M., Gehr, T., Vechev, M.: Differentiable Abstract Interpretation for Provably Robust Neural Networks. In: Proceedings of the 35th International Conference on Machine Learning (ICML 2018). pp. 3578–3586. PMLR (2018)
14. Mitchell, T.M.: The Need for Biases in Learning Generalizations. Rutgers CS tech report CBM-TR-117 (1980)
15. de Palma, A.: oval-bab. `https://github.com/oval-group/oval-bab` (2021)
16. Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., Madry, A.: Adversarially Robust Generalization Requires More Data. In: Advances in Neural Information Processing Systems 31 (NeurIPS 2018) (2018)
17. Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C.: The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In: The 2011 International Joint Conference on Neural Networks. pp. 1453–1460. IEEE (2011)
18. Suresh, H., Guttag, J.: A Framework for Understanding Sources of Harm throughout the Machine Learning Life Cycle. In: Proceedings of the 1st ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization. pp. 1–9 (2021)
19. Tjeng, V., Xiao, K., Tedrake, R.: Evaluating Robustness of Neural Networks with Mixed Integer Programming. In: 7th International Conference on Learning Representations (ICLR 2019). pp. 1–21 (2019)
20. 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)

21. Zhang, H., Yu, Y., Jiao, J., Xing, E.P., Ghaoui, L.E., Jordan, M.I.: Theoretically Principled Trade-off between Robustness and Accuracy. In: Proceedings of the 36th International Conference on Machine Learning (ICML 2019). pp. 7472–7482. PMLR (2019)
22. Zhang, H., Weng, T.W., Chen, P.Y., Hsieh, C.J., Daniel, L.: Efficient Neural Network Robustness Certification with General Activation Functions. Advances in Neural Information Processing Systems 31 (NeurIPS 2018) pp. 4939–4948 (2018)
23. Zhang, X., Evans, D.: Cost-Sensitive Robustness against Adversarial Examples. In: 7th International Conference on Learning Representations, ICLR 2019 (2019)

A Overview of used networks

	mnist-net				mnist-net_256x4				mnist-relu_4_1024			
	Recall		Number of instances		Recall		Number of instances		Recall		Number of instances	
Class	training	testing	training	testing	training	testing	training	testing	training	testing	training	testing
0	0.998	0.990	100	100	0.996	0.988	100	99	0.994	0.993	100	100
1	0.997	0.988	100	99	0.995	0.991	100	100	1.000	0.997	100	100
2	0.996	0.982	100	96	0.987	0.969	99	95	0.906	0.893	81	94
3	0.993	0.975	100	99	0.981	0.971	97	98	0.994	0.992	98	99
4	0.996	0.968	100	95	0.993	0.978	100	98	0.941	0.931	99	93
5	0.997	0.975	100	95	0.993	0.981	98	97	0.967	0.948	91	93
6	0.998	0.978	99	97	0.997	0.982	99	98	0.914	0.906	92	85
7	0.997	0.984	100	100	0.996	0.977	99	99	0.987	0.978	99	98
8	0.995	0.971	100	96	0.989	0.967	100	95	0.816	0.829	83	83
9	0.993	0.968	97	98	0.981	0.959	96	95	0.915	0.908	86	90
	0.996	0.996	996	975	0.991	0.990	988	974	0.945	0.941	929	935

Table 1: The number of images per class considered for verification and the testing and training accuracy overall testing and training instances for the 3 conventionally trained, fully connected ReLU networks. We include the number of correctly classified images per network per original class (indicated by recall) and the training and testing accuracy over all MNIST training and testing data, respectively. Note that the performance scores in this table were calculated over more than the number of verified instances.

B Kolmogorov-Smirnov tests

B.1 Robustness distributions aggregated

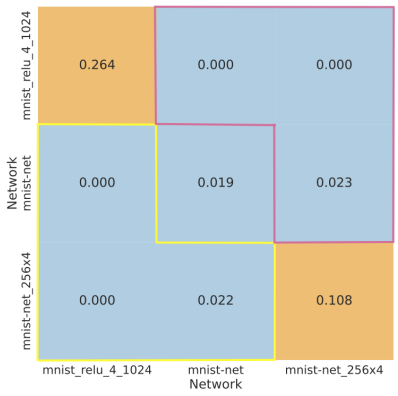
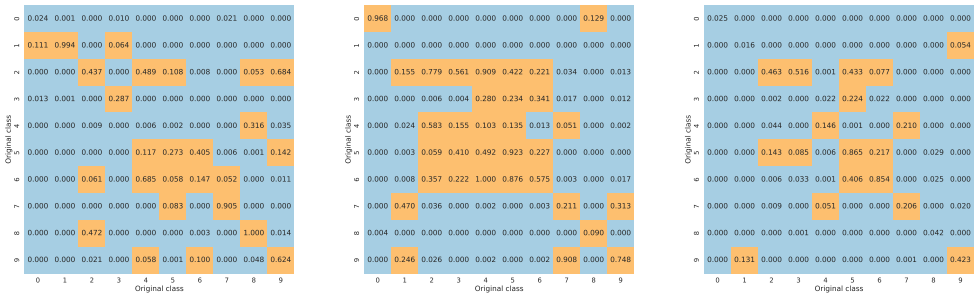


Fig. 8: Matrix containing the Kolmogorov-Smirnov test statistics with $\alpha = 0.05$ for the aggregated robustness distributions shown in Section 4.1. Each square contains the test statistic for the comparison of two robustness distributions. The squares coloured blue indicate that there is a significant difference in the robustness distributions according to the test and the yellow squares mean no significant difference can be proven. The squares in the lower half, outlined by a bright yellow, are the distributions created from the $\tilde{\varepsilon}^*$ of training data. The squares in the upper-half, outlined in pink, are the distributions created from the $\tilde{\varepsilon}^*$ of testing data. The squares on the diagonal, from upper-left to lower-right, are the distributions created from the $\tilde{\varepsilon}^*$ testing and training data from the same network. This way of displaying the test statistics is chosen for conciseness and the other matrices in this section have the same layout, albeit without the outlines.

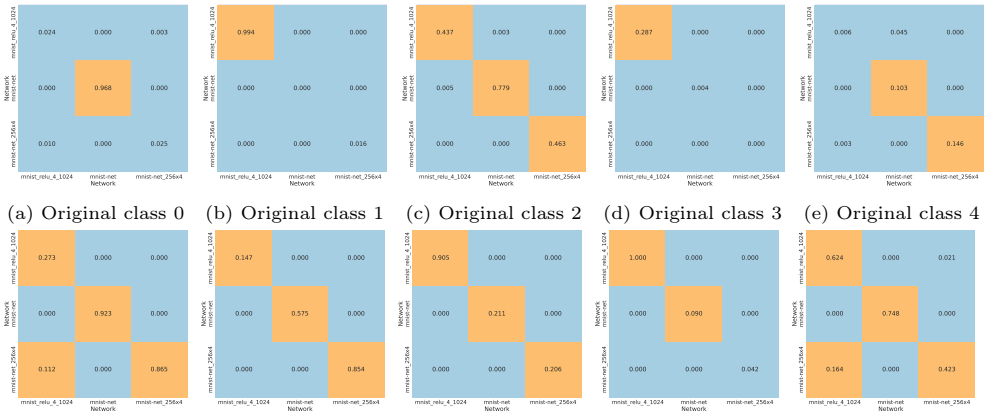
B.2 One-to-any KS statistics per network



(a) mnist_relu_4_1024 (b) mnist-net (c) mnist-net_256x4

Fig. 9: Matrices containing the Kolmogorov-Smirnov test statistics with $\alpha = 0.05$ for the one-to-any robustness distributions shown in Section 4.2. Each square contains the test statistic for the comparison of two robustness distributions of different original classes for the same network.

B.3 One-to-any KS statistics per class

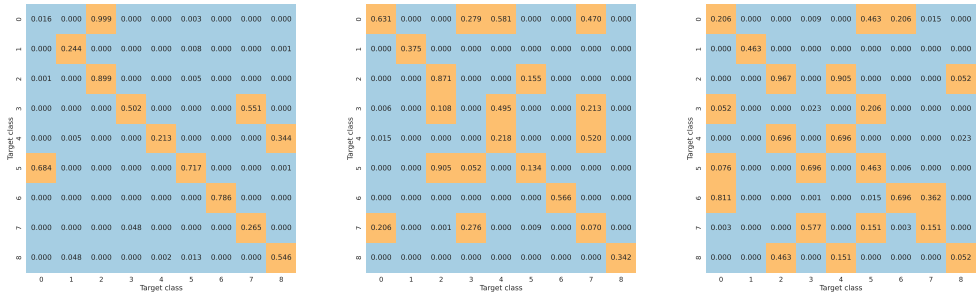


(a) Original class 0 (b) Original class 1 (c) Original class 2 (d) Original class 3 (e) Original class 4

(f) Original class 5 (g) Original class 6 (h) Original class 7 (i) Original class 8 (j) Original class 9

Fig. 10: Matrices containing the Kolmogorov-Smirnov test statistics with $\alpha = 0.05$ for the one-to-any robustness distributions shown in Section 4.2. Each square contains the test statistic for the comparison of two robustness distributions of the same orginal classes for the different networks.

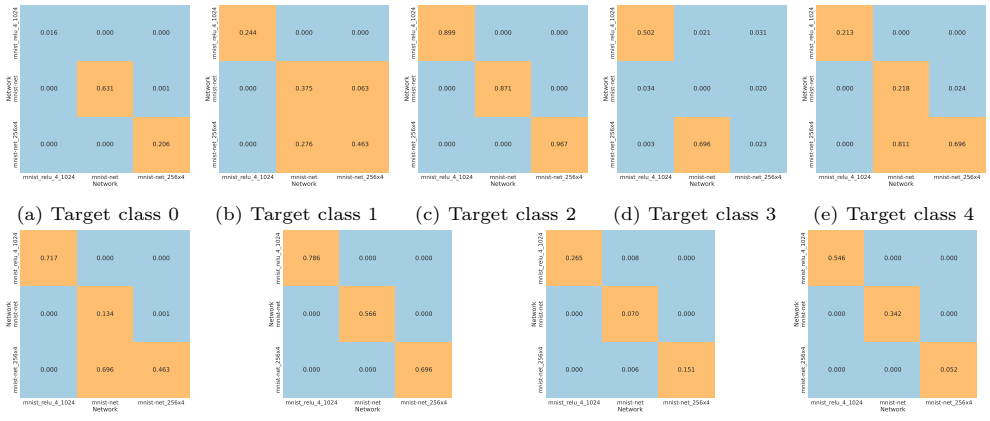
B.4 One-to-one KS statistics per network



(a) mnist_relu_4_1024 (b) mnist-net (c) mnist-net_256x4

Fig. 11: Matrices containing the Kolmogorov-Smirnov test statistics with $\alpha = 0.05$ for the one-to-one robustness distributions shown in Section 4.3. Each square contains the test statistic for the comparison of two robustness distributions of different target classes for the same network.

B.5 One-to-one KS statistics per class



(a) Target class 0 (b) Target class 1 (c) Target class 2 (d) Target class 3 (e) Target class 4

(f) Target class 5 (g) Target class 6 (h) Target class 7 (i) Target class 8

Fig. 12: Matrices containing the Kolmogorov-Smirnov test statistics with $\alpha = 0.05$ for the one-to-one robustness distributions shown in Section 4.3. Each square contains the test statistic for the comparison of two robustness distributions of the same target classes for the different networks.