SMART: A Robustness Evaluation Framework for Neural Networks

Yuanchun Xiong[1] and Baowen Zhang[1,2(✉)]

[1] Institute of Cyber Science and Technology, Shanghai Jiao Tong University, Shanghai 200240, China
pandada@sjtu.edu.cn

[2] Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, Shanghai 200240, China
zhangbw@sjtu.edu.cn

Abstract. Robustness is urgently needed when neural network models are deployed under adversarial environments. Typically, a model learns to separate data points into different classes while training. A more robust model is more resistant to small perturbations within the local microsphere space of a given data point. In this paper, we try to measure the model's robustness from the perspective of data separability. We propose a modified data separability index Mahalanobis Distance-based Separability Index (MDSI), and present a new robustness evaluation framework Separability in Matrix-form for Adversarial Robustness of neTwork (SMART). Specifically, we use multiple attacks to find adversarial inputs, and incorporate them with clean data points. We use MDSI to evaluate the separability of the new dataset with correct labels and the model's prediction, and then compute a SMART score to show the model's robustness. Compared with existing robustness measurement, our framework builds up a connection between data separability and the model's robustness, showing openness, scalability, and pluggability in architecture. The effectiveness of our method is verified in experiments.

Keywords: Neural network robustness · Data separability · Adversarial inputs · MDSI · SMART

1 Introduction

Recent work has demonstrated that neural networks (NNs) are vulnerable to adversarial examples: visually imperceptible perturbations that can mislead a well-trained model [1]. Safety is always a relative concept under adversarial environments. [2] suggests that the existence of adversarial examples is an inevitable part of the network architecture and an inherent weakness of network models. It is necessary to measure and improve the robustness of different models to

This work is supported by the National Key Research and Development Program of China under Grant No. 2020YFB1807504 and No. 2020YFB1807500.

M. Tanveer et al. (Eds.): ICONIP 2022, CCIS 1791, pp. 288–299, 2023.
https://doi.org/10.1007/978-981-99-1639-9_24

adversarial examples and mitigate the risks caused by the adversary. Currently, many robustness measurements focus on the success rate of attacks and the minimal distortion to successfully generate adversarial examples, few consider data separability when measuring robustness.

Separability is an intrinsic characteristic of a dataset. It describes how data points of different labeled classes are mixed together. Typically, a model learns to classify data points into different categories while training. The separability difference between the original dataset and the adversarial dataset can reflect the performance of models trained on them. [3] created a model-agnostic separability index called Distance-based Separability Index (DSI). It uses Euclid distance as its distance metric to measure how far data points are from each other.

In this work, we modify DSI and apply data separability to robustness evaluation. First, we propose Mahalanobis Distance-based Separability Index (MDSI), a modification of DSI that uses Mahalanobis distance as its metric and considers the correlation between different dimensions of a dataset when measuring separability. For a given dataset, we use different attack techniques to general some adversarial examples and mix them with the original clean examples to form a new dataset. The new dataset with correct labels and model's predicted labels show difference in separability. We use MDSI to reflect the difference and construct our robustness evaluation framework termed Separability in Matrix-form for Adversarial Robustness of neTwork (SMART). We highlight our main contributions in this paper as follows:

- We propose a data separability metric called MDSI. It shows the overall separability of a given dataset. In practice, we us e partitioned matrix operations to optimize the efficiency of computing MDSI.
- We introduce SMART, a robustness evaluation framework for neural network models. SMART measures model's robustness by comparing MDSI results on a new dataset consisting of clean data points and adversarial examples against the model. Our framework is scalable and shows flexibility in the choice of attacks and the proportion of adversarial examples generated.
- We use SMART and some mainstream metrics to evaluate the robustness of several state-of-the-art NN models. The results verify the effectiveness of our SMART framework.

2 Related Work

2.1 Adversarial Examples

A counter-intuitive property of neural networks found by [1] is the existence of adversarial examples, a hardly perceptible perturbation to a clean image can cause misclassification. [4] observes that the direction of perturbation matters most and proposes the Fast Gradient Sign Method (FGSM) to generate adversarial examples. Basic Iteration Method (BIM) [5] is an extension of FGSM by applying a smaller step size. Jacobian Saliency Map-based Attack (JSMA) [6]

only modifies a limited amount of pixels of input to search for adversarial examples. DeepFool [7] uses geometry concepts as its guide for search. C&W attack [8] formulates finding adversarial examples as a distance minimization problem and can find adversary with a significantly smaller perturbed distance.

On the opposite side of adversarial attacks, many defense techniques have been proposed to identify or reduce adversarial examples. There is an arms race between attacks and defenses. Adversarial training can improve robustness by retraining the model on adversarial examples [4]. It is by far the strongest empirical defense. There is no defense technique that is effective to all attacks.

2.2 Robustness Evaluation

Adversarial robustness is defined as the performance of a neural network model facing adversarial examples [9]. Some research formalizes their notion of robustness by giving their own definitions, including point-wise robustness [10], local robustness [11] and categorial robustness [12]. The core is when input changes within a small range, the output of a robust model shouldn't show large fluctuation. The evaluation of robustness can be achieved from different perspectives.

Accuracy. Model's accuracy on adversarial examples is the direct indicator of robustness. Model's accuracy on clean examples also reflects its performance and generalization. For convenience, we refer to the latter one as Nature Accuracy (NA).

Minimal Distortion Radius. The minimal distortion radius represents the range of adversarial perturbation for generating successful adversarial examples. In general, a model with a larger radius suggests higher adversarial robustness.

An upper bound of the radius is usually computed via some attacks and its tightness depends on the effectiveness of the attack [8]. We name the upper bound found by PGD attack as Empirical Radius (ER).

A lower bound is usually provided by certified methods, it guagrantees that the model is robust to any perturbations smaller than it. [13] proposed an attack-independent robustness metric CLEVER to gain a lower bound. However, [14] pointed out that gradient masking can misguide CLEVER to overestimation. Later discussion demonstrated that CLEVER can handle gradient masking problems [15].

2.3 Separability Index

Separability is an inherent characteristic of a dataset which measures the relationship between classes. [3] created Distance-based Separability Index (DSI) as a novel separability measure. It represents the universal relations between the data points in a dataset. A higher DSI score indicates that the dataset is easier to separate into different classes. When the DSI score is close to zero, it means that different classes of data have nearly identical distribution and are the most difficult to separate.

3 Method

In 3.1, we discuss about the relationship between model's robustness and data separability. On the basis of previous work on DSI mentioned in 2.3, we introduce a modified separability measure named MDSI in 3.2. In 3.3, we apply data separability to model's robustness evaluation and present our robustness evaluation framework SMART.

3.1 Model's Robustness and Data Separability

Given a NN model and its input space $\mathcal{X}$, let δ be the minimal distortion required to craft an adversarial example x' from a clean one x. Larger δ indicates that model is more robust around x. Consider the following two scenarios.

- Scenario I: In $\mathcal{X}$, only a few data points have relatively small δs.
- Scenario II: In $\mathcal{X}$, many data points have relatively moderate δs.

Robustness evaluation metrics that seek the bound of the minimal distortion radius focus on a single data point in the dataset at a time. These metrics often find models in Scenario I less robust because they have smaller minimal distortion radius. It is questionable because models in Scenario I generally work well when these few unrobust points are filtered out, while models in scenario II need to be strenghened at many unrobust locations before deployment.

Our method explores a novel approach based on data separability that simultaneously considers all data points in the dataset when evaluating robustness, and can reflect the overall robustness of neural network models.

3.2 The Separability Index MDSI

We propose Mahalanobis Distance-based Separability Index (MDSI) as a modification of DSI mentioned in 2.3. Intuitively, MDSI uses Mahalanobis distance as its distance metric, which has wide applications in image processing [19] and neurocomputing [18] areas.

Mahalanobis distance is unitless, scale-invariant, and takes the correlations of the dataset into account [16], and can better reflect the overall data separability when applied in MDSI. It requires to pass through all variables in the dataset to compute the underlying inter-correlation structure, so it is usually computationally more expensive than Euclidean distance [17].

Following are the steps to compute MDSI. Given a dataset $\mathcal{X}$ and two points $p, q \in \mathcal{X}$, let S be the covariance matrix of the dataset, d_M be the Mahalanobis distance between p and q:

$$d_M = d_{M(p,q)} = \sqrt{(p-q)^T S^{-1} (p-q)} \tag{1}$$

First, consider two classes $\mathcal{X}_m, \mathcal{X}_n \subset \mathcal{X}$ that satisfy $\mathcal{X}_m \bigcap \mathcal{X}_n = \emptyset$, they have the same distribution and sufficient data points. The Intra-class Mahalanobis Distance (IMD_m) set contains d_M between any two points in the same class $\mathcal{X}_m$.

$$\text{IMD}_m = \{d_{M(x_i,x_j)} \mid x_i, x_j \in \mathcal{X}_m \; ; x_i \neq x_j\} \tag{2}$$

The Between-class Mahalanobis Distance ($\mathrm{BMD}_{m,n}$) set contains d_M between any two points that are from different classes $\mathcal{X}_m, \mathcal{X}_n$.

$$\mathrm{BMD}_{m,n} = \{d_{M(x_i,x_j)} \mid x_i \in \mathcal{X}_m \; ; x_j \in \mathcal{X}_n\} \tag{3}$$

The Kolmogorov-Smirnov (KS) test quantifies a distance between the empirical distribution functions of two samples. Compared with other data distribution measures like Kullback-Leibler divergence, Jensen-Shannon divergence, and Wasserstein distance, KS test works when the samples have different number of points and is more sensitive when measuring separability [3].

MDSI uses KS test to examine the similarity of the distributions of IMD and BMD sets. Consider a n-class dataset $\mathcal{X}$, its subset $\mathcal{X}_i, \mathcal{X}_{\bar{i}}$ satisfies $\mathcal{X}_i \bigcap \mathcal{X}_{\bar{i}} = \emptyset, \mathcal{X}_i \bigcup \mathcal{X}_{\bar{i}} = \mathcal{X}$. The MDSI score of $\mathcal{X}$ is defined as:

$$\mathrm{MDSI}(\mathcal{X}) = \frac{1}{n} \sum_{i=1}^{n} \mathrm{KS}(\mathrm{IMD}_i, \mathrm{BMD}_{i\bar{i}}) \tag{4}$$

$\mathcal{X}$ has the lowest separability when the distributions of the IMD and BMD sets are nearly the same, and it shows the lowest MDSI score.

Advantages of MDSI. We using the *sklearn.datasets.make_blobs* function in Python to create eight two-class and five-class datasets and compare their DSI [3] and MDSI. Each dataset has 1000 data points and one cluster center per class, the Standard Variation (SD) of the clusters is set between 1 and 8. The results are shown in Fig. 1.

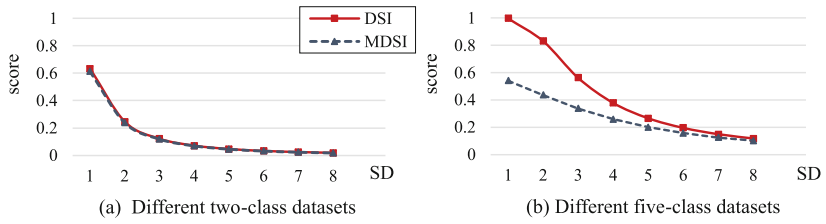

Fig. 1. DSI and MDSI scores on n-class datasets with different SDs.

As SD increases, the distributions of different classes overlap more and more, DSI and MDSI show the same downward trend, which is in line with our perception. The curves in (a) almost overlap, because the dimension is low and the correlation between dimensions is not obvious. In (b), the curves of MDSI is lower, indicating that when the class increases, the effect of dimensional correlation begins to appear, and the separability exhibited by MDSI is more realistic.

The comparison of the two experimental results shows that when the feature dimension of the dataset increases, the correlation between them has a greater impact on MDSI, and the difference between DSI and MDSI is more obvious. We consider MDSI to be a better separability metric because it takes into account the

influence of the dimensional correlation of the dataset and can more realistically reflect the overall data separability.

Here are some optimizations we made to the calculation of MDSI. The time cost for computing IMD and BMD sets increases quadratically with the number of data points. [3] encountered a similar problem and suggested that random sampling can reduce time cost without significantly affecting the results. However, we think their approach has its inherent defect. We optimize the computation of IMD, BMD, and MDSI by introducing partitioned matrix operation.

Small Batch. First, we apply the idea of training a neural network in small batches to the generation of BMD sets. Choosing an appropriate batch size can avoid out-of-memory issues no matter how large the dataset grows.

Partitioned Matrix Operation. By converting raw data points into matrices, we can use CUDA to speed up matrix operations and reduce the time cost. The operation is shown in Fig. 2. $\mathcal{X}_i$ is a class of dataset $\mathcal{X}$ where $\mathcal{X}_i \bigcap \mathcal{X}_{\bar{i}} = \emptyset, \mathcal{X}_i \bigcup \mathcal{X}_{\bar{i}} = \mathcal{X}$. The number of data points in $\mathcal{X}_i, \mathcal{X}_{\bar{i}}$ are M and N. The feature dimension of each point is F. When the covariance matrix S of the dataset is not full rank, it will be replaced by its pseudo-inverse matrix. $P = P_{M\times F} = (P_1, \ldots, P_M)^T$ and $Q = Q_{N\times F} = (Q_1, \ldots, Q_N)^T$ are two input matrices, each row represents a data point.

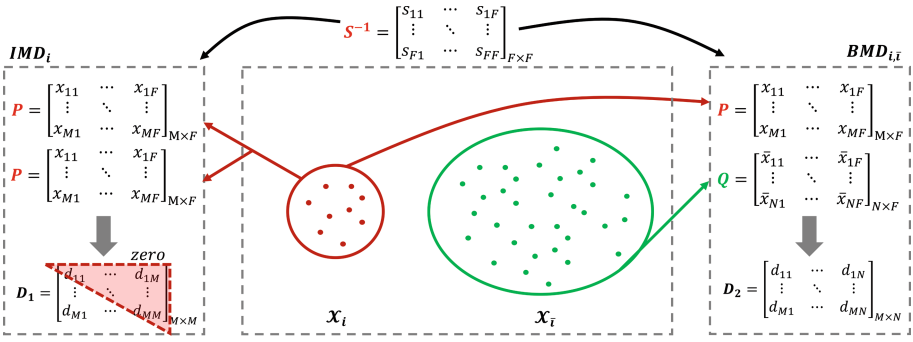

Fig. 2. The matrix operations in computation of IMD and BMD sets.

Take any row P_i and Q_j for example, their Mahalanobis distance is

$$d_{M(P_i,Q_j)} = \sqrt{P_i S^{-1} P_i^T - Q_j S^{-1} P_i^T - P_i S^{-1} Q_j^T + Q_j S^{-1} Q_j^T} \tag{5}$$

There are four matrix multiplication operations in the above formula. Extend the above formula to all data points, we get the distance matrix $D_2 = \mathrm{BMD}_{i\bar{i}}$ of size $M \times N$. In a distance matrix, each element d_{ij} represents a distance. D_2 can still be regarded as a combination of four matrix multiplication operations.

$$D_2 = \sqrt{M_1 - (QS^{-1}P^T)^T - PS^{-1}Q^T + M_2} \tag{6}$$

In Formula 6, M_1 and M_2 are two $M \times N$ matrices. First take out the diagonal elements of the $M \times M$ matrix $PS^{-1}P^T$ and get a $M \times 1$ vector, then replicate and extend it to a $M \times N$ matrix M_1. Similarly, replicate the $N \times 1$ vector consisting of the diagonal elements of matrix $QS^{-1}Q^T$, extend it to a $N \times M$ matrix which is the transpose of M_2. All elements in D_2 form the $\text{BMD}_{i\bar{i}}$ set.

The distance matrix $D_1 = \text{IMD}_i = \sqrt{2 \cdot M_3 - 2 \cdot PS^{-1}P^T}$ is a symmetric matrix, only elements in its strictly lower triangular matrix is needed to form the IMD_i set. M_3 can be obtained in the similar way as M_1 and M_2.

When the size of the input matrices P or Q is too large, we use the combination of small batch and matrix operation (i.e. partitioned matrix operation) for optimization. The above operations still apply to partitioned matrices.

We verified our optimization on Google Colab and the results show significant improvement. Computing MDSI on MNIST is almost 200 times faster, the calculation time reduced from 3387.79 s to 17.76 s. When we set partition size to 5000, the calculation time is 17.88 s, almost the same. The results indicate that matrix operation is far more efficient and small batch can solve the out-of-memory problem without significantly affect performance. For convenience, the default partition size is set to 10000, and partition matrix operations are applied when more samples are added.

3.3 The Robustness Evaluation Framework SMART

In this section, we combine MDSI and neural network models. We evaluate the model's robustness by measuring the separability difference between the datasets with correct labels and with model predicted labels.

Figure 3 shows the evaluation process for our framework SMART. We combine the standard and adversarial test sets into a new dataset. The score MDSI_0 of the new dataset with correct labels is considered as the separability reference result. The score MDSI_1 of the new dataset with model predicted labels is considered as the separability measurement result. We can use these two MDSIs to calculate the final SMART score that represents the model's robustness.

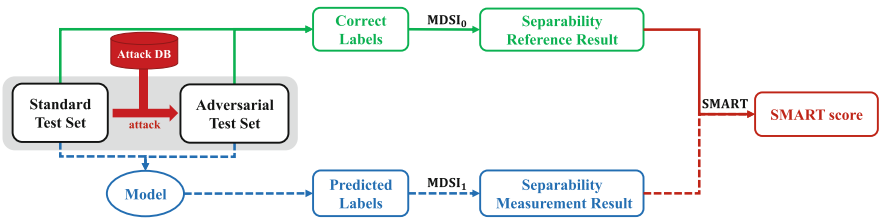

Fig. 3. The robustness evaluation framework SMART.

Attack DB. A flexible and critical component of our framework is the Attack Database (Attack DB). The idea is to put some typical attacks in the DB and

mix the generated adversarial examples with clean examples in appropriate proportions. In practice, FGSM [4], BIM [5], PGD [2], DeepFool [7] and C&W [8] are selected to join the Attack DB.

The proportion K of adversarial examples generated by different attacks in attack database A is basically an empirical parameter, which can be tuned by researchers using SMART. We think attacks with high time complexity should generate fewer samples. In practice, we use the time T for the attacks in A to generate the same number of adversarial examples on the same dataset as a reference for time complexity, and use it to determine the proportion K. Different $a_i, a_j \in A$ may vary widely in $t_i, t_j \in T$, so the relationship between $k_i, k_j \in K$ is determined by $k_i/k_j = \log t_j / \log t_i$.

SMART Formula. We expect to create a formula that utilizes the difference of the separability reference result MDSI_0 and measurement result MDSI_1 to reflect robustness. Through observation, we preset the following three formulas:

$$y_1 = 2 - \frac{\mathrm{MDSI}_1}{\mathrm{MDSI}_0},\ y_2 = \tanh(y_1),\ y_3 = \mathrm{Sigmoid}(y_1) = \frac{1}{1+e^{-y_1}} \tag{7}$$

Intuitively, higher SMART score represents a more robust model. We experimented with the above formulas using the configuration in 4.1. Their curves are in line with the expected trend, suggesting their validity in representing robustness. Among them, y_3 is more sensitive to changes and its results are normalized. We determine the final SMART score as $y = y_3 = \mathrm{Sigmoid}(2 - \mathrm{MDSI}_1/\mathrm{MDSI}_0)$ and present the results in 4.1.

Algorithm 1 summarizes the process of calculating SMART scores.

Algorithm 1: SMART score

Input: Dataset X and corresponding label C, model f, attack database A and proportion K, total number of attacks n, SMART formula y.

Output: SMART score ρ.

$\rho = 0$;
calculate the model's predicted labels $Y = f(X)$ on the clean dataset X;
for $i \leftarrow 1$ **to** n **do**
  use $a_i \in A$ to generate corresponding adversarial examples $X_i' = a_i(f, X)$;
  `/* |⋆| represents the total number of elements in set ⋆ */`
  use $k_i \in K$ to randomly choose $X_i'' \subseteq X_i'$ that satisfies $|X_i''| = k_i \cdot |X_i'|$;
  calculate the predictions $Y_i'' = f(X_i'')$ and correct labels C_i'' on X_i'';
  combine X, X_i'' and compute their separability MDSI_0 under labels C, C_i'';
  combine X, X_i'' and compute their separability MDSI_1 under labels Y, Y_i'';
  $\rho = \rho + y(MDSI_0,\ MDSI_1)$;
  i=i+1;
end
return $\rho = \rho/n$

4 Experiments

In this section, we make some experiments to demonstrate the sensitivity and validity of SMART in 4.1, and compare SMART with existing robustness evaluation metrics in 4.2.

For evaluation purposes, we implemented Algorithm 1 as a proof-of-concept tool, which is written in Python 3.8 and uses the PyTorch frameworks. All experiments mentioned in this section were run on the Google Colab environment.

4.1 The Validity of SMART

The upper limit of perturbation ϵ is set between 0 and 1 in increments of $\Delta = 0.1$. We compute the SMART scores of an AlexNet pre-trained on MNIST (nature accuracy 99.19%) under different ϵ and present the results in Table 1.

Table 1. The SMART scores of a pre-trained AlexNet under different ϵ

ϵ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
SMART	0.721	0.626	0.550	0.527	0.494	0.469	0.464	0.459	0.457	0.433

A larger ϵ indicates that larger perturbations may appear, and the probability of misclassification will increase accordingly. For the same model, when the perturbation gradually increases, it will appear to be less robust. As shown in Table 1, the SMART score of the AlexNet decreases as ϵ increases, which verifies the validity of SMART.

When ϵ is fixed, a more robust model will have more similar MDSI_0 and MDSI_1 with its SMART score closer to 1, a less robust model will have a SMART score farther from 1. SMART is more sensitive when $\epsilon \leq 0.5$.

4.2 SMART and Mainstream Robustness Metrics

We further experiment on the MNIST and CIFAR-10 (CIFAR for short) datasets, comparing SMART and mainstream robustness metrics, including Natural Accuracy (NA), Empirical Radius (ER), and CLEVER score mentioned in 2.2.

Table 2. Robustness evaluation results on MNIST.

	NA(%)		ER($\times 10^{-4}$)		CLEVER		SMART	
	Std	**Adv**	**Std**	**Adv**	**Std**	**Adv**	**Std**	**Adv**
LeNet-5	99.22	99.00	1.202	1.467	0.185	0.237	0.583	0.728
AlexNet	99.29	98.85	1.192	1.232	0.319	0.362	0.689	0.726

ER represents the upper bound of the minimal perturbation computed by attacks under the l_2 norm. CLEVER is set according to the original paper [8] where the sampling parameters batch size $N_b = 500$, the number of samples per batch $N_s = 1024$, the maximum perturbation $R = 2$ under l_2 norm and 100 test-set images for CIFAR and MNIST.

On MNIST, we evaluate these metrics on relatively small models Lenet-5 and AlexNet. Each model has a standard trained version (Std) and an adversarial trained version (Adv). The Adv models are enhenced via PGD [2] adversarial training with $\epsilon = 0.3$, $\alpha = 0.1$, $iteration = 40$ and random initiation.

The evaluation results are shown in Table 2. The change in natural accuracy shows that adversarial training slightly reduces the generalization ability of the model. ER and CLEVER show that adversarial training indeed makes the model more robust, showing higher scores. Comparing the SMART scores of the Std and Adv columns, the results show that the robustness of both models is improved after PGD adversarial training.

Table 3. Robustness evaluation results on CIFAR-10.

	NA(%)		ER		CLEVER		SMART	
	Std	**Adv**	**Std**	**Adv**	**Std**	**Adv**	**Std**	**Adv**
LeNet-5	63.82	62.58	0.07255	0.07279	0.0726	0.0875	0.2560	0.2610
ResNet-18	78.49	72.14	0.07278	0.07283	0.0181	0.0466	0.2558	0.2830
SqueezeNet	79.8	77.25	0.07273	0.07294	0.013	0.0603	0.2429	0.2653
VGG-16	81.93	74.51	0.07266	0.07298	0.0118	0.2271	0.2533	0.2746
AlexNet	82.94	77.99	0.07277	0.07281	0.0718	0.1682	0.2552	0.2602
DenseNet-121	89.42	70.87	0.07285	0.07311	0.0409	0.1837	0.2507	0.2855

On CIFAR-10, we additionally evaluate four other models VGG-16, DenseNet-121, ResNet-18 and SqueezeNet, the results are shown in Table 3. Both ER and CLEVER show that adversarial training improves the robustness of the models, although the changes in ER are very slight. The SMART scores of the Adv models are higher than those of Std models, which can reflect the changes in robustness of a single model unfer different training methods. Comparing SMART scores between different models under the same training method, the results in the last two columns show that after adversarial training, DenseNet and ResNet are more robust.

The above experiments verify that SMART is a reliable robustness evaluation framework, which matches well with mainstream robustness metrics such as ER and CLEVER on various models. Now we discuss the advantages of SMART over these attack-based or certification metrics.

SMART and Attacks. In theory, adversarial attacks developed to search for anti-robust perturbations of models around data points can only optimize to some local minima. In a sense, the attack-based method ER can only achieve

partial guarantees. Thus, a holistic robustness evaluation method is expected to be developed to reflect a more comprehensive robustness distribution in the input space. Compared to adversarial attacks that seek the upper bound of local minimal perturbations, SMART exploits all the anti-robust perturbations found by tools in the Attack DB and reflects the overall robustness of neural networks.

SMART and Certifications. CLEVER is an attack-agnostic robustness metric to estimate a lower bound of the minimal perturbation, which transforms the robustness evaluation process into a local Lipschitz constant estimation problem and applies the extreme value theory to solve it. While certification methods such as CLEVER and randomized smoothing can provide lower bound guarantees, SMART can be used to measure the overall robustness explored by Attack DB. Its evaluation results for a single model are as effective as the mainstream robustness evaluation metrics, and can also well reflect the robustness differences between different models.

Moreover, many current and future adversarial methods can be plugged into our attack library, Attack DB, according to the evaluation process in Fig. 3. Our proposed data separability index MDSI enables reasonable integration of all generated adversarial data. Therefore, SMART can be used as an open and pluggable framework to evaluate robustness.

5 Conclusion

In this paper, we propose SMART, a novel robustness evaluation framework for NN models. The main advantages of SMART over mainstream robustness evaluation methods are: (i) we develope a data separability index MDSI, which allows SMART to evaluate robustness more stably and suitably from the perspective of the overall dataset separability; (ii) we use partitioned matrix operations to significantly reduce the computation time of SMART and fix the out-of-memory issue; (iii) the Attack DB in SMART is open to accommodate a wide variety of adversarial methods, which makes our framework expandable.

Currently, the applicability of SMART has been verified with extensive experiments on datasets including MNIST and CIFAR-10 and on models including LeNet-5, AlexNet, ResNet-18, SqueezeNet, VGG-16, and DenseNet-121. The results show that SMART scores match and outperform mainstream robustness metrics when evaluating both natural and defended models. We plan to extend our work to ImageNet in future work.

References

1. Szegedy, C., Zaremba, W., Sutskever, I., et al.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
2. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
3. Guan, S., Loew, M., Ko, H.: Data separability for neural network classifiers and the development of a separability index. arXiv preprint arXiv:2005.13120 (2020)

4. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
5. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533 (2016)
6. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: 2016 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 372–387. IEEE (2016)
7. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2574–2582 (2016)
8. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE symposium on security and privacy (SP), pp. 39–57. IEEE (2017)
9. Bai, T., Luo, J., Zhao, J.: Recent advances in understanding adversarial robustness of deep neural networks. arXiv preprint arXiv:2011.01539 (2020)
10. Bastani, O., Ioannou, Y., Lampropoulos, L., et al.: Measuring neural net robustness with constraints. arXiv preprint arXiv:1605.07262 (2016)
11. Huang, X., Kroening, D., Ruan, W., et al.: A survey of safety and trustworthiness of deep neural networks: verification, testing, adversarial attack and defence, and interpretability. Comput. Sci. Rev. **37**, 100270 (2020)
12. Levy, N., Katz, G.: Roma: a method for neural network robustness measurement and assessment. arXiv preprint arXiv:2110.11088 (2021)
13. Weng, T.W., Zhang, H., et al.: Evaluating the robustness of neural networks: an extreme value theory approach. arXiv preprint arXiv:1801.10578 (2018)
14. Goodfellow, I.: Gradient masking causes clever to overestimate adversarial perturbation size. arXiv preprint arXiv:1804.07870 (2018)
15. Weng, T.W., Zhang, H., Chen, P.Y., et al.: On extensions of clever: a neural network robustness evaluation algorithm. In: 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pp. 1159–1163. IEEE (2018)
16. McLachlan, G.J.: Mahalanobis distance. Resonance **4**(6), 20–26 (1999)
17. Ghorbani, H.: Mahalanobis distance and its application for detecting multivariate outliers. Facta Univ. Ser. Math. Inf. **34**(3), 583–95 (2019)
18. Haldar, N.A.H., Khan, F.A., Ali, A., Abbas, H.: Arrhythmia classification using mahalanobis distance based improved fuzzy c-means clustering for mobile health monitoring systems. Neurocomputing **220**, 221–235 (2017)
19. Zhang, Y., Du, B., Zhang, L., et al.: A low-rank and sparse matrix decomposition-based mahalanobis distance method for hyperspectral anomaly detection. IEEE Trans. Geosci. Remote Sens. **54**(3), 1376–1389 (2015)