



Adversarial robustness improvement for deep neural networks

Charis Eleftheriadis¹ · Andreas Symeonidis² · Panagiotis Katsaros¹

Received: 13 September 2023 / Revised: 22 January 2024 / Accepted: 6 February 2024 / Published online: 14 March 2024
© The Author(s) 2024

Abstract

Deep neural networks (DNNs) are key components for the implementation of autonomy in systems that operate in highly complex and unpredictable environments (self-driving cars, smart traffic systems, smart manufacturing, etc.). It is well known that DNNs are vulnerable to adversarial examples, i.e. minimal and usually imperceptible perturbations, applied to their inputs, leading to false predictions. This threat poses critical challenges, especially when DNNs are deployed in safety or security-critical systems, and renders as urgent the need for defences that can improve the trustworthiness of DNN functions. Adversarial training has proven effective in improving the robustness of DNNs against a wide range of adversarial perturbations. However, a general framework for adversarial defences is needed that will extend beyond a single-dimensional assessment of robustness improvement; it is essential to consider simultaneously several distance metrics and adversarial attack strategies. Using such an approach we report the results from extensive experimentation on adversarial defence methods that could improve DNNs resilience to adversarial threats. We wrap up by introducing a general adversarial training methodology, which, according to our experimental results, opens prospects for an holistic defence against a range of diverse types of adversarial perturbations.

Keywords Adversarial examples · Adversarial training · Neural network security · Adversarial robustness

1 Introduction

Adversarial examples can be generated by inducing carefully crafted noise into the input data of neural networks, so that perturbed data contextually lead the machine learning algorithm to misbehaviour. This effect is more easily evident into image classification problems, where the misbehaviour takes the form of image misclassification. We focus on adversarial examples within the scope of neural network classifiers, though our contributions can be adjusted and applied to other neural network applications as well.

An adversarial example x' based on an image x , which is classified with label l by a function f (neural network), is defined as follows: x' is derived by applying a minimal

perturbation to x , such that the image x' is classified by f with label $l' \neq l$. In other words, if $\|\dots\|_p$ is an L_p distance metric (e.g. L_1 , L_2 or L_∞) between images in the input domain of f and $x - x' = \eta$ denotes a possible perturbation, then x' is given as:

$$\begin{aligned} & \min_{x'} \|x - x'\|_p, \\ & \text{such that } f(x) = l, \\ & \quad f(x') = l', \\ & \quad l \neq l' \end{aligned} \quad (1)$$

To better understand and explain the cause of misclassification, we visualize the effects of adversarial examples, in relation with the decision boundary of a neural network. Let us call Task the ideal decision boundary (dashed line curve in Fig. 1a), for a given classification problem. When constructing an adversarial example x' , the attacker tries to “shift” some image x outside of the network’s decision boundary (solid line curve). This is achieved by finding the minimal perturbation $\min_{x'} \|x - x'\|_p$, such that the neural network model is more prone to misclassification, i.e. the smallest projection of x to an hyperplane space, where the gap between the model’s decision boundary and Task is as large as possible.

✉ Charis Eleftheriadis
celefther@csd.auth.gr

Andreas Symeonidis
symeonid@ece.auth.gr

Panagiotis Katsaros
katsaros@csd.auth.gr

¹ School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece

² School of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Thessaloniki, Greece

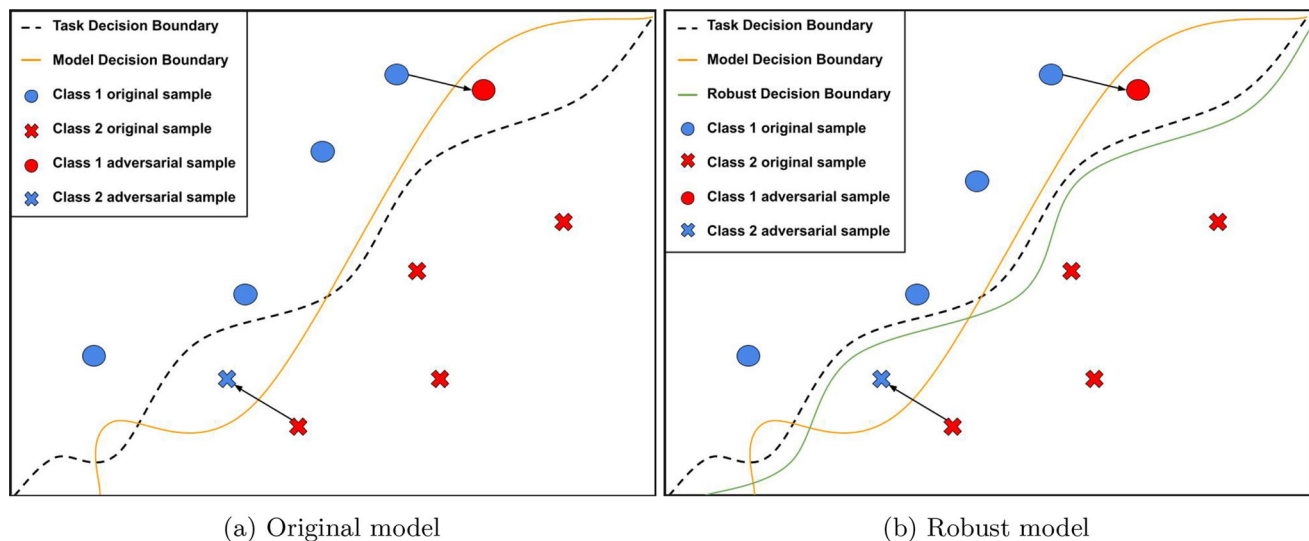


Fig. 1 Robustness of neural network models and the role of their decision boundary. A robust model limits the space that an attacker can exploit, for crafting adversarial samples

This vulnerability exposes the *robustness problem* of neural network models, where *robustness refers to how easy it is to find adversarial examples x' that are close to their original input x* . Moreover, it raises concerns about how safe and reliable can be the systems based on neural network components, given their robustness against such security threats. Practically, a robust model would reduce, as much as possible, the gap between its decision boundary and Task , thus limiting the attacker's capability to expose it into adversarial examples. In Fig. 1b, the robust model's decision boundary represented with green line separates the hyperplane space almost identically to Task , thus eliminating most possibilities for generating successful adversarial examples.

Two main facets of robustness are considered: attacker capabilities (attack methods) and defender capabilities (adversarial defence methods). Regarding the latter, we invest on improving *adversarial training*, as it seems to be the most promising and successful approach to achieve neural network robustness. This defence method aims to the direct exposure of the neural network, during training, to suitably selected perturbed data. As an outcome, the model forms a decision boundary that hopefully reduces the gap from Task (Fig. 1b).

Our motivation is to provide a practical framework for *holistic robustness improvement* by incorporating in adversarial training a representative ensemble of adversarial examples. Such a framework takes into account all characteristics of the various classes of adversarial examples identified in our taxonomy (Sect. 2.1).

More concretely, the contributions in this paper with respect to the attacker capabilities are:

- An extensive experimentation with a representative set of the most harmful adversarial attacks and a systematic evaluation of their effects (with respect to adversarial image quality, attack success rate, classification accuracy, confidence score and L_p distance metrics) on deep neural networks of varying complexity (convolutional and residual neural networks with varying numbers of parameters).
- An *Adversarial Robustness Evaluation Benchmark* (AREB), derived from our experimentation, capable to support an holistic evaluation of adversarial defence methods.

With respect to the defender capabilities, we rely on the hypothesis that a model trained with adversarial examples, suitably selected for representing all attack crafting strategies, is more resilient to diverse types of adversarial attacks that may arise in real-world scenarios. The model's ability to defend against such a multitude of attacks is enhanced through including adversarial examples that are specifically crafted to exploit diverse L_p norms, most notably L_1 , L_2 , and L_∞ . In light of this perspective, the two contributions of this work are:

- Experimental insight for the effectiveness of various input transformation methods (preprocess defences), which aim to leave less available ground for adversarial examples through “compressing” the models' input space.
- A road map comprised of techniques towards the holistic adversarial robustness improvement of deep neural networks and the evaluation of its effectiveness.

The outlined contributions support a process flow with concrete steps for adversarial robustness improvement, while the AREB benchmark provides a framework for the holistic evaluation of the defence methods that may combine adversarial training with preprocess defences.

In Sect. 2, we introduce a taxonomy of adversarial attacking methods based on their vital characteristics, as well as (experiments with) our Adversarial Robustness Evaluation Benchmark. In Sect. 3 we focus on the effectiveness of various preprocess defences on adversarial robustness and we provide experimental results for classifiers based on the MNIST and CIFAR-10 datasets. Section 4 introduces the adversarial training technique of our robustness improvement approach. The effectiveness of our techniques is assessed for the same datasets as the ones used in the experiments of the previous section. Moreover, additional adversarial robustness metrics are taken into account. Section 5 presents the steps of the process flow of our robustness improvement approach, along with the associated costs for applying it in terms of the needed human effort and computational resources. Finally, in Sect. 6 we wrap-up the exposition of the outlined contributions with insightful concluding remarks and suggestions for future research prospects.

2 Adversarial attacking methods

2.1 Taxonomy of adversarial attacking methods

Adversarial attacks comprise a wide range of techniques that aim to craft examples, with the goal to fool the machine

learning model under attack. In the related bibliography for attacks during test/inference time, the first paper on adversarial machine learning [1] introduced the notion of evasion attacks and a gradient-based algorithm. Nevertheless, the adversarial examples threat was not taken seriously until the moment of publication of the results in [2]. A countermeasure technique that was called adversarial training was introduced in [3], which made it possible to train robust models against adversarial attacks.

Although adversarial examples can be created for all types of machine learning models (decision trees [4], SVM [5]), more emphasis is given to neural networks, as these models exhibit top performance in various domains, including computer vision (image classification [6], object detection [7, 8]) and natural language processing [9, 10]. The first challenge in our research was to identify the most vital characteristics of adversarial attacks, so as to build a taxonomy of them. After having thoroughly surveyed multiple adversarial attacks, we concluded to the most dominant categories, in Table 1, along with the publication which is best known for each of them.

Two attack techniques are distinguished with respect to the *model knowledge*, namely the *white box* and *black box attacks*. In the former category, the attacker has complete access to any information required, for generating adversarial examples, like the model's architecture, the weights and the back-propagation derivatives, to name a few. On the contrary, in black box attacks, the attacker has limited access only to the input/output pairs (or generally the representation mapping between input and output) created by the model.

When adversarial attacks are classified with respect to the *attack target*, we distinguish between (i) a *targeted* attack,

Table 1 Taxonomy of adversarial attack methods **T**: Targeted, **NT**: Untargeted

Attack method	Model knowledge	Attack target	Attack strategy	L_p norm
Fast Gradient Sign Method (FGSM) [3]	White box	T/NT	Sensitivity Analysis	L_∞
Basic Iterative Method (BI) [11]	White box	T/NT	Sensitivity Analysis	L_∞
Iterative Least-Likely Class Method (ILC) [11]	White box	T	Sensitivity Analysis	L_∞
Projected Gradient Descent (PGD) [12]	White box	T/NT	Sensitivity Analysis	L_∞
Auto Projected Gradient Descent (APGD) [13]	White box	T/NT	Sensitivity Analysis	L_∞
Elastic Net Attack (EAD) [14]	White box	T/NT	Sensitivity Analysis	L_1
Jacobian Saliency Map Attack (JSMA) [15]	White box	T/NT	Sensitivity Analysis	-
Universal Adversarial Perturbation (UAP) [16]	White box	NT	Optimization	-
DeepFool [17]	White box	NT	Optimization	L_2
Carlini & Wagner L_2 Attack (CWL ₂) [18]	White box	T/NT	Optimization	L_2
Zeroth-Order Optimization (ZOO) [19]	Black box	T/NT	Optimization	L_2
Boundary Attack (BA) [20]	Black box	T/NT	Optimization	L_2
HopSkipJump Attack (HSJA) [21]	Black box	T/NT	Optimization	L_2
Sign-OPT Attack [22]	Black box	T/NT	Optimization	L_2
Square Attack [23]	Black box	T/NT	Optimization	L_2
Simple Black Box Attack (SimBA) [24]	Black box	T/NT	Optimization	L_2

where the model misclassifies a specific label given by the attacker and (ii) a *non-targeted* attack, in which the model misclassifies any label from the ones that it has been trained on.

The *attack strategy* deployed by the attacker is another discriminating characteristic that we took into account in our taxonomy. Two main aspects capture the nature of the attack strategy used [25]: (1) the type of perturbation (noise-based or geometric transformation) and (2) the class of algorithms/methods employed to find successful adversarial perturbations. Regarding the first aforementioned aspect, we focus on noise-based perturbation attacks, i.e. attacks that add white noise to carefully selected areas of an image (or more generally of any other input), as it is formulated in eq. (1). Robustness against natural transformations (e.g. rotations and translations) is beyond the scope of this work, and for this reason, Table 1 does not include attacks based on geometric transformations. Regarding the second aforementioned aspect, there are three main techniques used to generate adversarial examples:

- *Sensitivity Analysis*. This technique is based on algorithms for analysing the gradient of the loss function with respect to the input. The ultimate goal is to disclose the importance of a given feature (e.g. pixel) in the overall decision process.
- *Optimization*. According to this approach, the search of adversarial perturbations is performed using optimization algorithms and constraints.
- *Generative*. New adversarial examples are generated from the probability distribution of successful adversarial perturbations, which is captured through the use of various generative models (e.g. Variational Auto Encoders—VAEs [26] or Generative Adversarial Networks—GANs [27]).

The last mentioned category of techniques is not represented in Table 1, because the results in the related bibliography seem to be not very competitive compared to those derived from the use of the two other attack strategies. Moreover, the associated cost (time and resources) for applying such an approach seems to be significantly higher than the corresponding cost for any of the other techniques. Whereas this category falls outside the scope of our study, intriguing works, such as the one by [28], offer a systematic perspective on the phenomenon, accompanied by practical and efficient countermeasures.

Perturbations of adversarial attacks are quantified using various *norms*, i.e. functions that map vectors to non-negative scalars. The distance between two vectors is measured by the norm of their difference $\|x - x'\|_p$ that always returns a positive scalar. Four norms are usually used for adversarial attacks in the image classification domain:

- L_0 , which can be viewed as a cardinality function

$$\|x\|_0 = \#(i \mid x_i \neq 0) \tag{2}$$

for the features (pixels) of the original image that have been perturbed in an adversarial example.

- L_1 , also known as the Manhattan norm

$$\|x\|_1 = \sum_{i=1}^n |x_i| \tag{3}$$

measures the sum of magnitudes of the vectors in a given space.

- L_2 , also known as the Euclidean distance between two vectors x and x' :

$$\|x - x'\|_2 = \left(\sum_{i=1}^n |x_i - x'_i|^2 \right)^{\frac{1}{2}} \tag{4}$$

- L_∞ that is based on the maximum norm

$$\|x - x'\|_\infty = \max(|x_1 - x'_1|, \dots, |x_n - x'_n|) \tag{5}$$

for quantifying the maximum change to any coordinate. For images, this norm represents the maximum bound of change for each pixel, i.e. the number of pixels that are modified is not taken into account.

2.2 Physical-world attacks

The attack methods mentioned so far are applied on the pixel space of images. Another emerging surface of adversarial attacks is the so-called *physical-world attacks*. This research field is focused on how adversarial attacks can be launched in more realistic scenarios “in the wild”.

A first attempt of exposing state-of-the-art classifiers in such threats was reported in [11], where attacks against various images were created and then the original and the distorted images were printed in paper. Next, a picture of the printed images was taken using a cell phone camera, in order to evaluate the classifiers performance. The whole experimental context was realized indoors, in order to ensure stable conditions with respect to the lighting and brightness conditions, the distance and the angle with which the printed pictures were captured by the camera. Nevertheless, reductions in the accuracy of the investigated models compared to the digital images were noted, even for the printed samples that did not contain distorted images.

Furthermore, in [29] the authors considered the problem of crafting effective physical-world attacks, for varying physical conditions (distance, view angle, brightness, etc.),

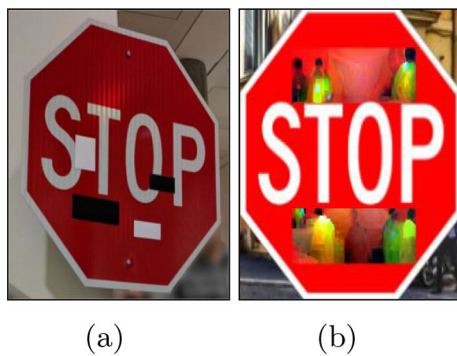


Fig. 2 Appropriate placement of stickers on stop signs, eventually leads state-of-the-art image classifiers (2a) and object detectors (2b) to unpredictable behaviour

by taking into account environmental factors and the fabrication error of the printing procedure. To this end, a methodology called *robust physical perturbations* (RP_2) was proposed, according to which printed stickers are placed strategically in appropriate spots of stop signs, as shown in Fig. 2a. In this manner, it was possible to fool some top image classifiers that are employed in autonomous driving applications. In [30], the authors have extended their experimental approach to object detectors. More specifically, they launched two kinds of physical-world attacks against state-of-the-art object detectors (YOLO [8], Faster R-CNN [7]), namely the *disappearance* and *creation* attacks, which resulted in reduced performance of the attacked models. An instance of a creation attack against the YOLO algorithm is depicted in Fig. 2b.

There are notable related works on physical-world adversarial attacks against systems with learning components, like for example the research reported in [31, 32]. Especially for safety-critical systems, it is essential to exclude potentially catastrophic behaviours and therefore the robustness of learning components against any kind of possible perturbation has to be shown. As a step towards this direction, in next sections we introduce a systematic process for holistically improving the models' robustness against a wide variety of adversarial attack methods. Still, the outcome of applying such an holistic robustness improvement procedure has to be evaluated with respect to the models' resilience against physical-world attacks, which is not within the scope of this article.

Finally, a noteworthy related work is the benchmark of perturbations in [33] that are relevant to the physical world, which in most cases are not adversarial in nature (minimum projection out of the decision boundary). That work is complementary to our benchmark, which is introduced in Sect. 2.3. In a related context, there are studies like [34] that systematically assess the relationship between adversarial examples and diverse types of noise perturbations. Their proposed framework integrates a Convolutional Denoising

Auto Encoder and a Classifier, demonstrating enhanced performance against a spectrum of adversarial attacks.

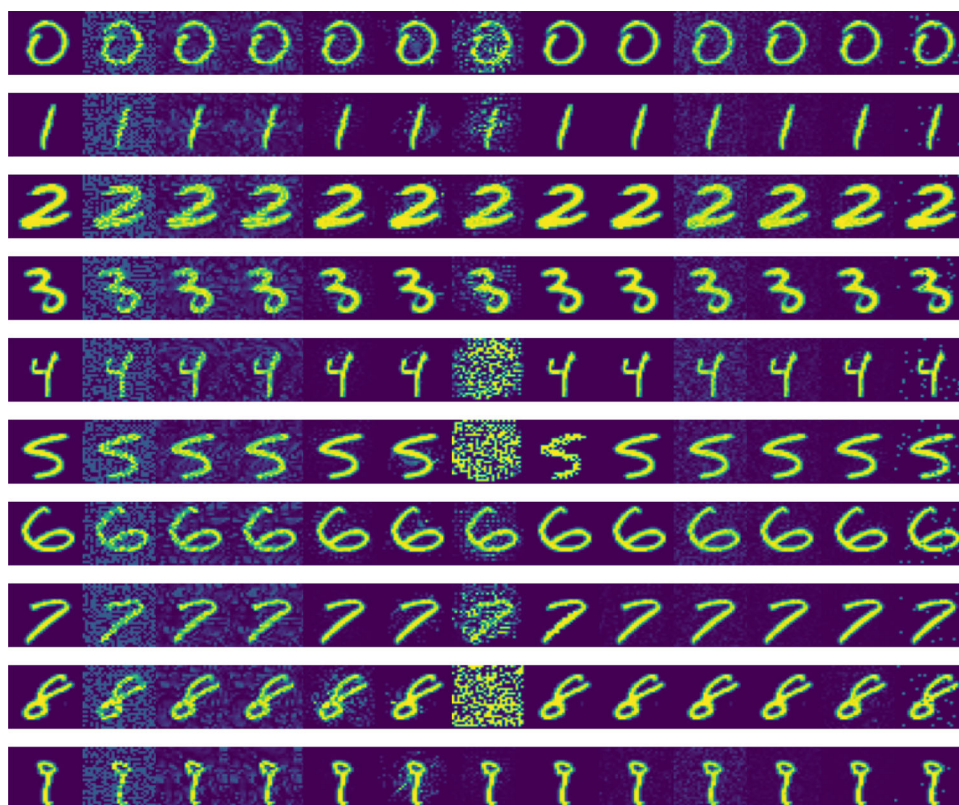
2.3 Adversarial robustness evaluation benchmark

The diverse conditions, under which experimental evaluation of adversarial machine learning takes place in related works, make it hard to compare any improvements in the robustness of neural networks. More concretely, a model's robustness is usually measured with respect to the adversarial attacks selected for its evaluation. If there are adversarial defences that were never tested, robustness will not reach an adequate level, while, on the other hand, robustness may be easily overestimated, if attacks that have not been taken into account can potentially bypass the defence mechanism employed.

This problem highlights the need for a general framework that could be justifiably considered as benchmark for adversarial robustness. A framework based on a systematic selection of attack crafting methods allows to test and evaluate a model's adversarial robustness, from an holistic point of view. A noteworthy methodology is the one in [35], which resulted in the AutoAttack benchmark [13], an ensemble consisting of four white box and black box attacks. According to the authors, their benchmark is more systematically built than the one in [36], which seems to have a narrow scope, and it is computationally more feasible than the benchmark in [37], which includes overly many attacks for the evaluation of adversarial robustness.

We introduce a new benchmark, called Adversarial Robustness Evaluation Benchmark (AREB), which supports a different approach from [35] for *testing models that are robust against multiple types of adversarial attacks*. AREB is based on the taxonomy of Table 1 and therefore consists of attacks that represent all diverse characteristics of adversarial examples, including both white box and black box attacks, as well as attacks based on all possible norms and attack strategies for adversarial perturbations. The effectiveness of each case of attack was evaluated in extensive experiments. The set of attacks in the AREB was selected by taking into account five criteria in order to find those that seem to have the biggest impact on neural networks of different complexity and size. Another difference of AREB from the benchmarks in [35] and [13] is that the latter support *only* improvements in the robustness against the worst-case perturbation out of the four attacks of the benchmark, as opposed to our approach, which takes into account *simultaneously* all the attacks of the AREB. In this way, we aim to address the fact that a model that happens to exhibit robust performance against a given type of attack, it may happen to be much less robust against another attack that leverages a different distance metric and/or attack strategy.

Fig. 3 Image Grid of Adversarial Examples on MNIST. From *top to bottom*, one sample per class (all digits (0 – 9)) is depicted. From *left to right*, commencing with the original image and progressing with the adversarial examples generated by diverse attacking methods. More precisely: *Original image, FGSM, BI, PGD, EAD, JSMA, DeepFool, CWL₂, ZOO, BA, HSJA, Sign-OPT and SimBA*



To justify the selection of attacks that are included in AREB, we note the following remarks in regard to the different characteristics considered in our taxonomy (Sect. 2.1).

First the vast majority of attacks in the related bibliography are White box attacks; therefore, Table 1 includes less Black box attacks. Moreover, all attacks based on Sensitivity Analysis are inevitably White box attacks, since a prerequisite for creating an adversarial example is being able to access the model. On the other hand, all Black box attacks are based on Optimization techniques. Sensitivity analysis techniques adopt a simpler and faster approach towards creating adversarial examples, and for this reason, they are preferred in adversarial training schemes. Optimization techniques are in general harder and more time-consuming, but they present better performance in finding small imperceptible adversarial perturbations.

The L_p norm exploited is clearly correlated with the Attack Strategy chosen: attacks that are based on Sensitivity Analysis exploit the L_1 and L_∞ norms, while those based on Optimization techniques use the L_2 norm. We do not see a specific technical reason behind this correlation, besides that these are the common choice of norms within the machine learning community when employing the Sensitivity Analysis and Optimization strategies. L_2 is commonly used as a regularization term in optimization problems to derive solutions that are smooth and have small magnitudes. The L_1 and L_∞ norms quantify the difference between data points,

respectively, in terms of the magnitude and the maximum budget of noise added. This renders them a natural choice for evaluating the sensitivity of a neural network's decision with respect to deviations from the input data distribution.

In image classification problems the work reported in [38] demonstrated that the L_p norm used for crafting adversarial examples, adds noise following particular patterns. These patterns eventually create image features that have to be learned, if we want to improve the robustness of models. We assume that a model trained on perturbations derived from all different norms and attack strategies, is more resilient to diverse types of adversarial attacks.

To ensure that all characteristics of attacks in our taxonomy (Table 1) are represented in the AREB, a minimum of four categories of adversarial attacks will have to be included, namely:

- White Box L_1 Sensitivity Analysis Attack
- White Box L_2 Optimization Attack
- White Box L_∞ Sensitivity Analysis Attack
- Black Box L_2 Optimization Attack

The AREB set of attacks includes the most effective attacks from each of the aforementioned categories that are selected based on the findings of the experiments in next subsections. These are the attacks with the highest misclassification rate for the minimum distortion (i.e. visual difference, smallest

perturbation). AREB is intended to serve as a benchmark for assessing the robustness of a neural network model against all types of attacks at the same time. Holistic robustness improvement means that the model demonstrates improved behaviour against all the four attacks in AREB. Since these attacks seem to be the most effective in their category, evaluating a model’s robustness against them shows the worst-case impact with respect to all other attacks in the same category. For instance, if a defence technique delivers models that are robust against PGD, then the model is supposed to exhibit at least similar, if not better, robustness on average against all other White Box L_∞ Sensitivity Analysis attacks (e.g. FGSM, BIM, ILC).

2.3.1 Criteria of effectiveness of adversarial attacks

We introduce the five criteria for evaluating the effectiveness of adversarial attacks and selecting those that are included in the AREB.

Visual difference The most dangerous attack crafting methods produce examples that do not affect the decision made by humans regarding the class shown in the images. If the noise induced is noticeable and it heavily deforms the original inputs, then the attack method is disqualified, even if it is effective with respect to the remaining criteria.

Attack success rate This metric refers to the percentage of input data, on which the attack method can be deployed effectively. If X is the set of input data for a neural network f and

$$X' = \{x' : x' = x - \eta \text{ for } x \in X \text{ and } \eta \text{ some perturbation}\}$$

let us call

$$X_{adv} = \{x' \in X' : \text{equation (1) is true}\}$$

i.e. the labels of elements in X_{adv} differ from the labels computed by f , for the original data $x \in X$. Then, the attack success rate is defined as

$$\frac{|X_{adv}|}{|X'|} \% , \text{ where } |S| \text{ denotes the cardinality of } S$$

Classification accuracy This metric quantifies the difference of a model’s performance before and after the attack has been deployed. Let us denote by l_x the label of $x \in X$ and $l = f(x)$ the label computed by f . Then, the classification accuracy of the neural network f is given by

$$\frac{|x \in X : l_x = f(x)|}{|X|} \%$$

, whereas the classification accuracy for the adversarial examples is given by

$$\frac{|\{x' = x - \eta : f(x') = l_x \text{ for } x' \in X'\}|}{|X'|} \%$$

Average confidence score Even when the label computed by a neural network is correct, the confidence quantifies the degree of certainty for the decision made. The range of confidence values is from 0 (no confidence) to 1.0 (full confidence). Let us consider a binary classification problem and two neural networks with the *softmax* function used as activation function of the output layer. The prediction made by these two models will be in the form of a 2-size vector with the probabilities computed for each of the two classes summing up to 1, e.g. [0.53, 0.47] and [0.98, 0.02]. In this example, while both models predict the first label as correct, the confidence of their decisions differs significantly (0.53 and 0.98). For a set of test samples, we are interested to measure the average confidence score over them.

Median L_p distance metrics The L_p metrics quantify in the pixel space the distance between the pixels of the original and the adversarial images. Low L_p distance values show “small” perturbations and indicate a more effective attack, if they achieve their target. We use the L_0 , L_2 and L_∞ distance metrics, and for a set of test samples, we measure their median L_p distance. These metrics are interpreted as follows. L_0 measures the percentage of pixels of original samples altered. L_2 quantifies the change in the pixel values and L_∞ the percentage of noise added to any pixel in the worst case.

2.3.2 Experiments on the effectiveness of adversarial attacks and selection of AREB attacks

For each one of the attacks in Table 1, we have taken into account its risk based on the visual difference of adversarial examples and we have then evaluated its effectiveness (attack success rate, classification accuracy, etc.) if it represents an actual threat. The set of attacks in Table 2 that comprise the AREB seem to be real threats that were found to be the most effective in their category. At the same time, these attacks represent all diverse characteristics of our taxonomy in Table 1. *Experimental framework* The values of attack parameters in our experimental context were selected as follows. For the FGSM and PGD attacks, we used the values proposed in the original papers. For all other attacks, the values causing the *highest attack success rate, for a minimal perturbation* were selected. These values were found by preliminary experimentation. This approach is fundamentally different from [16], which advocates to find a fixed image-agnostic perturbation vector causing label changes for most images sampled from a data distribution. Such a perturbation vector would be inde-

Fig. 4 Image Grid of Adversarial Examples on CIFAR-10. From *top to bottom*, one sample per class (order: *ship, deer, truck, airplane, frog, cat, automobile, dog, horse, bird*) is depicted. From *left to right*, the order is identical with the one presented in Fig. 3. More precisely: *Original image, FGSM, BI, PGD, EAD, JSMA, DeepFool, CWL₂, ZOO, BA, HSJA, Sign-OPT and SimBA*



pendent of the attack method and the L_p norm selected and would not fit into the attack categories selected for the AREB.

We focused exclusively on the untargeted attack scenario that seems a more realistic setting given the complexity of the classification problems (MNIST, CIFAR-10) in our experiments. This choice directly excludes the ILC attack [11], which would be appropriate for a targeted attack evaluation scenario, for datasets with much more classes that are not highly distinguishable (e.g. ImageNet [39]). However, the experimental evaluation for a third dataset was left as future research.

For evaluating the attack risk based on the visual difference criterion, we visually inspected multiple images derived from adversarial perturbations, as the ones shown in Figs. 3 and 4. In most cases, the adversarial examples are visually identical to the original images, but for a few cases of the MNIST dataset (e.g. second column of image grid in Fig. 3—FGSM Attack) the added noise was visible. This was

not observed in images with slightly higher resolution (e.g. CIFAR-10 cases), except of a single instance for all attacking methods (7th column of the image grid in Fig. 4—DeepFool Attack) where we observed occasionally very noisy examples. According to these findings, adversarial examples are in most cases an *invisible threat*, especially in the domain of image classification.

For the remaining evaluation criteria, 100 test samples were randomly selected from each dataset, for which we applied adversarial attacks on both the CNN and ResNet neural network architectures. From these experiments, we report here only the results obtained for the ResNet architectures, since all other results for the CNN architectures do not exhibit noteworthy differences. Although it is occasionally mentioned in the literature that the more complex and “deeper” architectures can be a soothing factor for the attack effectiveness, we found that this is not true, since deep neural networks like ResNet are equally vulnerable to adversarial attacks, as

Table 2 Adversarial robustness evaluation benchmark (AREB)

Attack methods	Model knowledge	Attack strategy	Lp norm
Projected gradient descent (PGD)	White box	Sensitivity analysis	L_∞
Carlini & Wagner L_2 Attack (CWL ₂)	White box	Optimization	L_2
Elastic net attack (EAD)	White box	Sensitivity analysis	L_1
HopSkipJump Attack (HSJA)	Black box	Optimization	L_2

any other architecture. A more detailed description of the experimental framework is given in Appendix 1, whereas the attack parameters used are provided in Appendix 1.

Results for adversarial attacks The experimental results concerning the ResNet models for the MNIST and the CIFAR-10 datasets are shown, respectively, in Tables 3 and 4. The last column in both tables displays the time (in seconds) taken to successfully generate 100 adversarial examples for each attack method.

As it is shown, all attacks exhibit nearly the highest possible success rate ($\sim 100\%$). No neural network architecture is adequate without any other robustness improvement measure, even if it is the relatively more deep and complex ResNet architecture. Additionally, it is noteworthy that in some cases the average confidence score for the adversarial examples is higher than that for the original samples, indicating that the model under attack may be more confident for the (wrong) predictions, compared to the behaviour for the original data, when the model is not attacked.

A matter of interest is also the extent to which an attack impacts the original image. As it is shown in Table 3, the PGD attack alters the image pixels to a relatively high extent (74.88% for L_0) with a moderate intensity (30% for L_∞), while the Carlini & Wagner L_2 Attack alters the image pixels only up to 24.30% for L_0 with a much higher intensity (99.22% for L_∞). For the CIFAR images that are more complex (3 channels and larger in size, i.e. height \times width) than the MNIST images, we observe that a smaller perturbation suffices to achieve comparable results. Given that both datasets include tiny images, we can conjecture that for more realistic images similar to those found in everyday applications (e.g. image size $> 224 \times 224 \times 3$), even when a fraction of a pixel is altered, this may be enough to cause the model to misbehave, as reported in [40].

Next we describe in detail each of the four attacks that comprise the AREB and we explain the reason for having been included in the benchmark.

Projected gradient descent (White Box L_∞ sensitivity analysis) attack The PGD attack [12] was originally based on the L_∞ norm, but it has been also applied using other L_p norms (i.e. L_1, L_2). PGD can produce very effective adversarial examples with limited computational cost that wrong predictions with high confidence. Moreover, it can be seen as an optimized version of the BIM attack: for every iteration towards finding the gradient descent's projection with the maximum impact, a step is made towards the direction of the negative loss function. As shown in Tables 3 and 4, this is the only attack that can achieve up to 100% success rate with average confidence score 1.0 and this is evident in both datasets. Also, as can be seen in Table 4, the ResNet model is more confident for the (wrong) predictions (1.0) for adversarial samples, than it is, for the original inputs (0.91). PGD demonstrates superior performance compared

to all other white box L_∞ sensitivity attacks, across all evaluation criteria in both datasets. Therefore it is our choice in this category of adversarial attacks.

Carlini & Wagner L_2 (White Box L_2 Optimization) attack. Carlini & Wagner (CW) attack variants [18] have shown very high success rates, and they are therefore considered among the most powerful adversarial attacks. In this type of attacks, the main goal is to find, through a binary search, the optimal value for a constant that is added in the objective function to be maximized for computing the adversarial sample. These attacks highlight the efficiency of the so-called iterative attacks, which are based on optimization methods. In Tables 3 and 4 we observe an attack success rate of 100% for both datasets and very low classification accuracy for adversarial samples. In all experiments, we applied the L_2 -based version of the attack. The original sample is barely altered by the attack, thus resulting in adversarial examples that as shown in Table 4 differ with respect to the median L_2 and L_∞ metrics by only 0.12 and 1.54%, respectively. Compared to the other attack of this category, i.e. the DeepFool, CWL_2 is more effective with respect to all evaluation criteria. Its main drawback is the relatively high computational costs (Table 3—2340s, Table 4—3447s) for crafting adversarial examples, something that is observed in most attacks based on optimization techniques (e.g. ZOO, SignOPT Attack, SimBA). With DeepFool, on the other hand, we can craft adversarial examples really fast (Table 3—67s, Table 4—34s), but this is achieved at the expense of their quality. More specifically, there are many occasions where DeepFool introduces a significant amount of noise into the original image (see 7th column of the image grids in Figs. 3 and 4), resulting in perturbed examples that cannot be considered as *adversarial* with respect to the visual difference criterion.

Elastic Net (White Box L_1 Optimization) attack.

The EAD attack can be seen as an amalgam of the CW attack method combined with the elastic-net regularization technique [14, 41] that is widely used in solving high-dimensional feature selection problems. It aims to generate examples as similar as possible with the original image, through “penalising” the adversarial samples that differ significantly. EAD is the only attack of our taxonomy (see Table 1) that is originally based on the L_1 norm. The results in Tables 3, 4 highlight its effectiveness compared to the other white box attacks under all evaluation criteria.

HopSkipJump (Black Box L_2 Optimization) attack. The HSJA attack [21] is an optimized version of the Boundary Attack [20]. The principal “improvement” lies in the algorithm's capability to generate adversarial samples of superior quality (reduced distortion) within a shorter timeframe (fewer queries by the search algorithm). Tables 3 and 4 demonstrate that all black box attacks exhibit are particularly effective, even though the information available to the

Table 3 Attack evaluation on MNIST (ResNet) **SA**: Sensitivity Analysis Attack, **O**: Optimization Attack

Attack method	Attack Category	Attack Success Rate (%)	Classification Accuracy adversarial (original)	Avg. Confidence Score adversarial (original)	Median distance metrics		Time (s)	
					L_0 (%)	L_2		L_∞ (%)
Fast Gradient Sign Method (FGSM)	White box L_∞ SA	95	5%(100%)	0.93(1.00)	58.16	6.1	30	0.26
Basic Iterative Method (BI)	White box L_∞ SA	97	3%(100%)	0.94(1.00)	75.32	4.0	30	9.21
Projected Gradient Descent (PGD)	White box L_∞ SA	100	0%(100%)	1.00(1.00)	74.88	3.8	30	9.26
Auto-PGD CE	White box L_∞ SA	100	0%(100%)	1.00(1.00)	84.76	5.7	30	14
Auto-PGD DLR	White box L_∞ SA	98	1%(100%)	0.99(1.00)	98.02	5.4	30	17
Elastic Net Attack (EAD)	White box L_1 SA	100	0%(100%)	1.00(1.00)	51.66	1.6	55.15	1218
Jacobian Saliency Map Attack (JSMA)	White box	100	0%(100%)	0.44(1.00)	5.42	2.6	99.98	339
DeepFool	White box L_2 O	97	8%(100%)	0.82(1.00)	58.67	7.3	99.97	67
Carlini & Wagner L_2 Attack (CWL₂)	White box L_2 O	100	0%(100%)	0.89(1.00)	24.30	6.2	99.22	2340
Zeroth-Order Optimization (ZOO)	Black box L_2 O	98	5%(100%)	0.81(1.00)	78.14	3.7	26.45	3788
Boundary Attack (BA)	Black box L_2 O	98	2%(100%)	0.54(1.00)	83.61	1.8	23.51	64
HopSkipJump Attack (HSJA)	Black box L_2 O	99	1%(100%)	0.93(1.00)	84.95	1.2	19.57	41
Sign-OPT Attack	Black box L_2 O	98	5%(100%)	0.50(1.00)	64.41	0.8	16.67	9989
Square Attack	Black box L_2 O	98	4%(100%)	0.73(1.00)	21.11	1.8	31.76	231
Simple Black Box Attack (SimBA)	Black box L_2 O	95	5%(100%)	0.58(1.00)	3.76	2.7	50	279

Bold indicates the most dominant Adversarial Attacks per category as a result of the attack evaluation performed. These instances constitute the AREB set, as presented in Table 2

Table 4 Attack evaluation on CIFAR-10 (ResNet) SA: Sensitivity Analysis Attack, O: Optimization Attack

Attack method	Attack Category	Attack Success Rate (%)	Classification Accuracy adversarial (original)	Avg. Confidence Score adversarial (original)	Median distance metrics		Time (s)	
					L_0 (%)	L_∞ (%)		
Fast Gradient Sign Method (FGSM)	White box L_∞ SA	96	14% (89%)	0.81(0.91)	100	1.6	3	0.3
Basic Iterative Method (BI)	White box L_∞ SA	98	11% (89%)	0.82(0.91)	100	1.2	3	1.7
Projected Gradient Descent (PGD)	White box L_∞ SA	100	9% (89%)	1.00(0.91)	100	1.1	3	2.1
Auto-PGD CE	White box L_∞ SA	100	9% (89%)	1.00(0.91)	100	1.2	3	2.8
Auto-PGD DLR	White box L_∞ SA	100	12% (89%)	0.74(0.91)	100	1.3	3	3.6
Elastic Net Attack (EAD)	White box L_1 SA	100	8% (89%)	0.79(0.91)	36.83	0.12	3.12	1370
Jacobian Saliency Map Attack (JSMA)	White box	99	9% (89%)	0.41(0.91)	2.38	0.5	18	452
DeepFool	White box L_2 O	100	9% (89%)	0.69(0.91)	100	1.8	26.45	34
Carlini & Wagner L_2 Attack (CWL₂)	White box L_2 O	100	8% (89%)	0.73(0.91)	100	0.12	1.54	3446
Zeroth-Order Optimization (ZOO)	Black box L_2 O	98	11% (89%)	0.47(0.91)	88.26	1.3	17.73	3788
Boundary Attack (BA)	Black box L_2 O	100	8% (89%)	0.50(0.91)	100	0.15	12.51	49
HopSkipJump Attack (HSJA)	Black box L_2 O	100	7% (89%)	0.90(0.91)	100	0.1	10.77	57
Sign-OPT Attack	Black box L_2 O	98	10% (89%)	0.50(0.91)	37.63	0.5	8.12	11802
Square Attack	Black box L_2 O	96	11% (89%)	0.59(0.91)	48.44	1.9	26.31	483
Simple Black Box Attack (SimBA)	Black box L_2 O	97	11% (89%)	0.54(0.91)	2.2	0.8	10	453

Bold indicates the most dominant Adversarial Attacks per category as a result of the attack evaluation performed. These instances constitute the AREB set, as presented in Table 2

attacker is limited and takes the form of input–output pairs from the trained neural network. This finding is reflected in the very high attack success rates and the very low values of classification accuracy. While in most evaluation criteria the performance of black box attacks is nearly identical, there are two criteria in which HSJA dominates. These are the Average Adversarial Confidence Score (Table 3—0.93, Table 4—0.90) and the drastically reduced computational cost (Table 3—41 s, Table 4—57 s).

3 Preprocess defences

A preprocess defence transforms a model’s inputs so that it will be hard for an attacker to exploit any of them. Most of these methods—except of label smoothing—lie on “compressing” the input space, for limiting the available space where the generation of adversarial examples is feasible. The robustness of neural network models is thus indirectly enhanced, while it is possible to combine this defence with adversarial training. In this section, we report the experimental findings for the most promising preprocess defences in two different contexts: when they are applied alone, as well as when they are combined with other preprocess defences.

3.1 Experimental framework

The effectiveness of preprocess defences was studied for both the CNN and the ResNet architectures, in the two datasets used in our experiments.

To find the parameter value(s) that best fit(s) to a preprocess defence, each method was applied in two different cases: original input data and adversarial input data. The rationale behind this is to ensure that any robustness gains are not accompanied by significant reductions in accuracy, for the original data, which has been observed in several studies of the related literature. Using diagrams like the one in Fig. 5, we found the optimal value for which a defence achieves the “best protection” for the model, while minimizing a possible reduction in accuracy. More specifically, Fig. 5 shows how the classification accuracy of a model is affected when increasing the hyperparameter lambda (λ) for the Total Variance Minimization (TVM) defence. The blue line refers to the model’s accuracy when TVM is applied to the original data, whereas the red line shows the model’s accuracy when the method is applied to adversarial data. In this particular case, we chose $\lambda = 0.09$, a value for which we observe that the model achieves 82% accuracy for the original data and almost 60% for the adversarial data.

To study the impact of preprocess defences on adversarial robustness, they were applied to the two datasets in this “optimal configuration”. This means that the accuracy in this preliminary tuning phase was not taken into account to judge

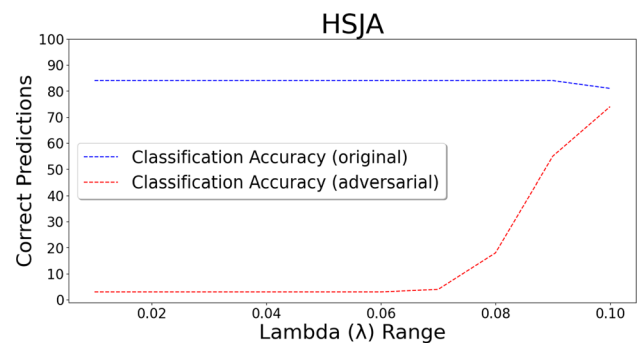


Fig. 5 Parameter tuning of Total Variance Minimization under the Hop-SkipJump attack

on the adversarial robustness of the models under test. The hyperparameters that were eventually chosen are specified in Appendix 2.

3.2 Experimental results

Tables 5 and 6 summarize the results for the preprocess defences under test, when we apply to the subject model two very effective attacks of the AREB, namely PGD and HSJA, a white box (sensitivity analysis) attack and a black box (optimization) attack. No noticeable differences were found in the parameter values of the preprocess “optimal configurations” in the two aforementioned attack cases. Therefore, we used the same hyperparameter value for both attack methods that were tested. On the other hand, we found significant differences in the parameter values, when the same defence method was applied to different data sets; for this reason we report the related results in separate tables.

Based on the classification accuracy shown for a model when it is and when it not under attack, the effectiveness of preprocess defences is evident for the white box attack. In the case of the black box attack, we do not observe the same improvement. Furthermore, certain defences (e.g. Feature Squeezing) do not have any impact at all in the case of the CIFAR-10 dataset, whereas for the MNIST dataset we notice significant differences, for all defence methods. The results for the ResNet models are omitted; in most cases—except for Label Smoothing—they are inferior than the corresponding results for the CNN models.

Finally, we also conducted experiments for testing all possible combinations of preprocess defences. Some of the results for the CIFAR-10 dataset are shown in Table 7. Only the combinations of defences that included Label Smoothing showed promising improvements. As a general observation, it seems that the model’s accuracy for the original data deteriorates when the number of defence methods that are combined is increased.

Hereafter, the results of Tables 5, 6 and 7 are commented separately, for all preprocess defences under test.

Table 5 Preprocess defence evaluation on MNIST (CNN) under adversarial attacks

Preprocess defence	Attack Methods	
	Projected Gradient Descent	HopSkipJump Attack
Label Smoothing [42]	100% (100%)	1% (100%)
Feature Squeezing [43]	79% (100%)	0% (100%)
Spatial Smoothing [43]	25% (100%)	2% (100%)
JPEG Compression [44, 45]	0% (98%)	4% (100%)
Total Variance Minimization [46]	7% (96%)	3% (100%)

For every preprocess defence applied to the model, we report the classification accuracy when this “shielded” model is under attack. The values in parenthesis correspond to the classification accuracy of the corresponding model when deployed in original data

Table 6 Preprocess defence evaluation on CIFAR-10 (CNN) under adversarial attacks

Preprocess defence	Attack Methods	
	Projected Gradient Descent	HopSkipJump Attack
Label Smoothing [42]	64% (84%)	8% (84%)
Feature Squeezing [43]	0% (83%)	8% (83%)
Spatial Smoothing [43]	26% (81%)	4% (81%)
JPEG Compression [44, 45]	1% (84%)	7% (84%)
Total Variance Minimization [46]	0% (82%)	6% (82%)

For every preprocess defence applied to the model, we report the classification accuracy when this “shielded” model is under attack. The values in parenthesis correspond to the classification accuracy of the corresponding model when deployed in original data

Table 7 Evaluation of combined preprocess defences on CIFAR-10 (CNN) under adversarial attacks

Preprocess defences combination	Attack Methods	
	Projected Gradient Descent	HopSkipJump Attack
FS + JC	0% (81%)	4% (81%)
FS + TVM	0% (83%)	9% (83%)
FS + SS	0% (77%)	8% (77%)
FS + LS	58% (83%)	4% (83%)
SS + TVM	7% (81%)	3% (81%)
SS + JC	10% (72%)	9% (72%)
SS + LS	62% (81%)	7% (81%)
JC + TVM	4% (77%)	5% (77%)
JC + LS	48% (78%)	6% (78%)
TVM + LS	62% (82%)	5% (82%)
FS + SS + LS	57% (77%)	6% (77%)
FS + JC + LS	45% (81%)	3% (81%)
SS + JC + LS	45% (72%)	14% (72%)
SS + JC + TVM	8% (79%)	8% (79%)
FS + SS + TVM	0% (77%)	6% (77%)
FS + JC + SS + LS	43% (74%)	6% (74%)
FS + JC + SS + TVM	8% (73%)	9% (73%)
FS + JC + SS + TVM + LS	46% (73%)	5% (73%)

FS: Feature Squeezing, **SS:** Spatial Smoothing, **JC:** JPEG Compression, **LS:** Label Smoothing, **TVM:** Total Variance Minimization

Label Smoothing

This is a technique [42] for “smoothing” a model’s labels so that it will not be overly certain for its predictions. This is applied to the penultimate layer of the neural network, by shrinking the differences between the logit of the correct class and the logits of the incorrect classes. As shown in Table 5, this defence is the only one that has the potential to protect a model against the PGD. The same is also true in the case of the CIFAR-10 dataset (Table 6).

Spatial Smoothing

This technique [43] is an input transformation that applies a median filter, for smoothing all image pixels based on their nearby pixels. Such a transformation eventually reduces the overall variation of the pixels significantly. From the results shown in Tables 5 and 6, it seems that spatial smoothing may have only marginal impact, when it is used to defend against adversarial attacks.

Feature Squeezing

This is an input transformation technique [43] that reduces the bit depth of images, i.e. the number of bits used for the colour representation of each pixel. As a consequence, it is potentially harder to generate an adversarial image. From the results in Table 5, Feature Squeezing seems to be effective in the case of the MNIST dataset that includes 8-bit depth images, but it is not effective at all, for the CIFAR-10 dataset (Table 6) that includes 24-bit depth images.

JPEG Compression

This technique transforms the subject image by applying factorization to its matrix representation; the compression

ratio is negatively correlated with the resulting image quality. It has been proposed in [44, 45] as a means to restrain the success rate of adversarial attacks. However, our experiments suggest (and it has been also reported elsewhere) that the accuracy for the original input data deteriorates, for higher compression ratios. JPEG compression seems to be effective for low-distance perturbations of images, as opposed to bigger perturbations, for which it is not equally effective. In the results shown in Tables 5 and 6, it seems not to be particularly effective, since they refer to attacks with relatively high success rates that mostly involve comparatively bigger perturbations.

Total Variance Minimization (TVM)

According to [46], the total variance of pixel values may be minimized when altering selected pixels randomly through Bernoulli sampling using a parametrizable function. From the experimental results in Tables 5 and 6, TVM seems not to be particularly effective for our attack settings, but it exhibits a decent effectiveness when it is combined with Label Smoothing (Table 7).

4 Adversarial training

Adversarial training was introduced in [3] as an approach with the potential to build robust models against adversarial attacks. Through including adversarial samples in training data it was observed that the model could classify to some extent correctly the perturbed inputs. Such an approach can be implemented through merging into a single loss function the original objectives of the model with the adversarial objectives; for an input x with label y the enhanced loss function is given as

$$\tilde{J}(\theta, x, y) = \alpha J(\theta, x, y) + (1 - \alpha) J(\theta, x + \epsilon \operatorname{sign}(\nabla_x J(\theta, x, y))), \quad (6)$$

where $J(\theta, x, y)$ denotes the loss function used in training with model parameters θ , α is the ratio of original to adversarial inputs during training, and ϵ is a parameter that adjusts the perturbation magnitude. The adversarial crafting method that is depicted on the right-hand side of equation (6) is the FGSM.

In [12], this training method was generalized to other attack methods as well by adding an optimisation perspective to the aforementioned scheme. Thus, adversarial training took the form of a robust optimisation problem that consists an *inner maximization* problem and an *outer minimization* problem; for $x \in \mathbb{R}$ with corresponding labels $y \in [k]$ these two problems are defined as

$$\min_{\theta} \rho(\theta), \quad \text{where} \quad \rho(\theta) = \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\max_{\delta \in \mathcal{S}} J(\theta, x + \delta, y) \right] \quad (7)$$

with \mathcal{D} representing the data distribution (number of items per feature) and δ quantifying the applied perturbation depending on the L_p norm used ($\mathcal{S} \subseteq \mathbb{R}$ denotes the set of allowed perturbations). The inner maximization problem yields the “strongest” possible adversary (δ value) for x that maximizes the loss, whereas the outer minimization problem expresses the need to minimize the loss function, given this specific adversary. In essence, finding the adversarial perturbation that maximises the loss function is a matter of optimizing the benefits of adversarial training, in terms of the robustness gains to be achieved.

Targeting to an holistic improvement approach against all attacks in the AREB, we focus exclusively on ensemble adversarial training methods, which were first described in [47]. Our training strategy is accomplished through the integration of attacks in equation (7), in the form of properly selected δ perturbations. Two main approaches [48] have been considered:

- **“Max” strategy:** for all inputs, the model is trained in each epoch using only the “strongest” adversarial example crafted from n different attack methods
- **“Average” strategy:** for all inputs, the model is trained in each epoch using all adversarial examples crafted from n different attack methods.

To the best of our knowledge, [49] is the only related work that aims to an holistic improvement of adversarial robustness. However, the adversarial training method in that work targets the worst-case over the union of adversarial perturbations (based on L_1 , L_2 and L_∞ norms), i.e. they follow a slightly different approach of the “max” strategy, as opposed to our approach that takes into account all types of perturbations at each training epoch (“average strategy”). Through this alternative approach, we increase the number of adversarial examples, on which the model is trained on. Moreover, our evaluation is based on the AREB benchmark, which is an outcome of systematic experimentation, whereas in [49] there is no systematic method for justifying the representativeness of the set of attacks selected. This can potentially result in overestimating the robustness improvement, if there are additional “powerful” attacks that have not been taken into account.

The research question for the experiments carried out on robustness improvement is:

Can we achieve resilience against all types of adversarial attacks, if we encompass in adversarial training through properly selected perturbations all the adversarial characteristics?

To answer this question, we have integrated into our ensemble adversarial training framework perturbations of all different L_p norms (L_1 , L_2 , L_∞) and attack strategies (Sensitivity Analysis, Optimization). Based on the experimental

results presented in Sect. 2.3.2, we opted to prioritize the selection of attack methods that maximize the loss function $J(\theta, x + \delta, y)$ in equation (7) compared to other adversaries. For this reason, the attacks selected are:

- the PGD attack [12], one of the most effective L_∞ -based attacks, which justifies why it is frequently used in most adversarial training solutions;
- the EAD attack, which has been found to be the most effective L_1 -based attack;
- the HSJ attack, which has been found to be the most powerful black box optimization-based L_2 attack.

4.1 Adversarial robustness metrics

In this section, we review the most widely used metrics for adversarial robustness and then we explain the rationale behind the selection of those used in our experimentation for answering the research question posed.

Empirical Robustness In [17], the so-called empirical robustness was proposed, for quantifying adversarial robustness. This metric is based on the L_2 norm and corresponds to the minimum needed perturbation to be introduced in the input data, so as the model will change its prediction.

CLEVER Another metric for adversarial robustness was proposed in [50], which is attack-agnostic. More specifically, by relying on Lipschitz continuity, the authors search for the local Lipschitz constant through applying extreme value theory. It is then possible to estimate the lower bound of the minimum adversarial distortion; any perturbation up to this bound corresponds to perturbed examples that are not adversarial ($f(x) = f(x') = l$). This metric is calculated per input sample, and it is known as the CLEVER metric.

ψ -metric based on KL Divergence In another proposal [51], adversarial robustness for a given perturbation range is quantified by the maximum divergence between the model's predictions for the original data and the worst-case adversarial example within the considered perturbation range. This approach is based on the Kullback–Leibler (KL) divergence metric, denoted as D_{KL} , which is a well-known way to measure the divergence of two probability distributions (in our case, prediction distributions). For an input sample x and a perturbation δ , the $\psi(x)$ metric for adversarial robustness is defined as:

$$\psi(x) = \frac{1}{\max_{\delta} D_{KL}(P(x), P(x + \delta))}, \quad (8)$$

where $P(x)$ denotes the prediction results for input x and $P(x + \delta)$ the results for the adversarial input. Typically, a low KL divergence value indicates high similarity between the prediction distributions of a model for the original and adversarial inputs. Therefore, for higher values of $\psi(x)$ the

robustness of a model against adversarial attacks is getting improved.

Loss Sensitivity

Although originally not proposed as a metric for assessing adversarial robustness, it could provide valuable insights into this topic. In [52], the metric was employed to evaluate the extent of memorization displayed by Neural Networks trained on real versus random input data. The metric simply measures the norm of the loss gradient with respect to input samples and is formally defined as:

$$g_x = \left\| \frac{\partial L}{\partial x} \right\|_1 \quad (9)$$

where g_x is the average norm of the loss gradient over a number of input samples x , referred as *loss sensitivity*. Higher metric values indicate a steeper average gradient over the input samples, while lower values suggest a loss gradient more resilient to changes concerning data that the associated model has not been trained on. Regarding adversarial robustness, lower metric values are connected with models exhibiting resistance to changes, as adversarial examples typically deviate from the data distribution on which the model was trained.

In regard with the research question posed, empirical robustness is not relevant, since the question refers to potential robustness improvement against all types of adversarial attacks. We remind that empirical robustness is based on the L_2 norm, whereas our AREB benchmark includes attacks that are based on L_1 and L_∞ metrics, as well. Moreover, as it was noted in Sect. 2.3.2, we have already configured the selected adversarial attacks with the parameters resulting in the minimum perturbation that can cause as high attack success rate as possible. Therefore, since empirical robustness refers to the minimum needed perturbation for the model to change its prediction, we do not expect that it can provide additional valuable information. To sum up, when focusing on the holistic improvement of adversarial robustness, like in our case, the CLEVER, ψ and loss sensitivity metrics are the most appropriate to assess it.

4.2 Experimental results

To answer the research question posed, we compare two “adversarially robust”, namely one with using only PGD in adversarial training and another one trained by combining the PGD, EAD and HSJA attacks. Both models were trained for 390 epochs and integrated the label smoothing preprocess defence, which seems to be less effective for the CIFAR-10 dataset than the MNIST dataset (especially in the case of the HSJA L_2 -based attack - cf. Sect. 3). The training took place using exclusively adversarial perturbations, as opposed to other approaches that mix adversarial

and original data in ratios 0.5 or more. This choice was preferred due to preliminary experimental findings suggesting that when using only adversarial perturbations we can achieve optimal results with respect to the model’s robustness, without undermining significantly the accuracy for the original data. The accuracy figures shown in the results give the average of 10 measurements taken from 100 test samples that each time were randomly selected among the test data.

Table 8 shows the classification accuracy of our adversarially trained neural networks, for the MNIST dataset, when they are attacked by all methods in the AREB. The second column refers to the accuracy of the adversarially trained neural network using the PGD L_∞ -based attack, while the third column shows the corresponding accuracy for the adversarially trained neural network using the PGD, EAD and HSJA attacks. Finally, the last table row shows the classification accuracy of the two adversarially trained neural networks, for the original MNIST data. It is evident that our approach for holistic improvement of adversarial robustness results in significant gains (shown in bold) with respect to the classification accuracy, for all attacks of the AREB. Moreover, the accuracy of the robust neural networks for the original data has been preserved.

Regarding the CIFAR-10 experimental results that are shown in Table 9, we observe a comparatively lower classification accuracy for the original data by both adversarially trained models. However, this drop is comparatively less than the drop observed for the original neural network, when applying Label Smoothing (cf. Table 6). It is again evident that when using the adversarially trained model with the ensemble of attacks (PGD/EAD/HSJA), the accuracy is

Table 8 Adversarial training on MNIST (CNN)

AREB	PGD Adv. Training [12] (%)	Robust Neural Network (%)	(PGD + HSJA + EAD)
Projected Gradient Descent (PGD)	87.3	98.4	
Carlini & Wagner L_2 Attack	14.0	58.6	
Elastic Net Attack (EAD)	12.6	27.5	
HopSkipJump Attack (HSJA)	10.1	20.8	
benign	97.0	99.0	

Bold indicates the improved performance of the proposed methodology compared to the standard method for increasing adversarial robustness Performance comparison for the two robust neural networks on AREB, where the classification accuracy is reported. Both models integrate Label Smoothing preprocess defence

Table 9 Adversarial training on CIFAR-10 (CNN)

AREB	PGD Adv. Training [12] (%)	Robust Neural Network (%)	(PGD + HSJA + EAD)
Projected Gradient Descent (PGD)	44.2	15.8	
Carlini & Wagner L_2 Attack	16.0	36.8	
Elastic Net Attack (EAD)	7.0	22.4	
HopSkipJump Attack (HSJA)	4.0	41.3	
benign	74.0	69.0	

Bold indicates the improved performance of the proposed methodology compared to the standard method for increasing adversarial robustness Performance comparison for the two robust neural networks on AREB, where the classification accuracy is reported. Both models integrate Label Smoothing preprocess defence

vastly improved, in all cases of the AREB, apart from the case of PGD attack. This phenomenon can be attributed to the limited impact of Label Smoothing on these specific data (CIFAR-10).

Table 10 summarizes the evaluation of adversarial robustness using the CLEVER metric. The figures shown provide estimates for the lower bound up to which perturbed examples are not adversarial, with respect to the L_p metric used by the adversarial attack. Thus, it is possible to compare the robustness of different models for the same attack or even to compare the robustness against attacks that share the same L_p metric. For each AREB attack, we kept the same parameter settings for CLEVER, with the ones used in the original paper that first introduced this method. The estimates shown for each model and attack are the average CLEVER value from 100 test samples. The second row provides the CLEVER values for the adversarially trained neural network using the PGD L_∞ -based attack, while the third row shows the corresponding values for the adversarially trained neural network using the ensemble of PGD, EAD and HSJA attacks.

As it is shown in Table 10, the average lower bounds given by the CLEVER metric are higher for the adversarially trained model with the combined PGD/EAD/HSJA attacks, in almost all cases. This implies that an attacker will have to introduce larger perturbations to force the model to misbehave; if more noise is needed for the attacker to achieve his goal, then the visual difference from the original sample is also magnified, which makes it more likely for the attack to be detected [53]. As noted in Tables 8, 9, when HSJA is included in adversarial training, a higher

Table 10 CLEVER values for the MNIST and CIFAR-10 datasets and three different models

NN model	AREB	L_p norm	CLEVER	
			MNIST	CIFAR-10
Original model	PGD	L_∞	0.003	0.007
	CWL ₂	L_2	0.082	0.017
	EAD	L_1	0.62	0.159
	HSJA	L_2	0.01	0.002
PGD Robust Model	PGD	L_∞	0.04	0.02
	CWL ₂	L_2	0.012	0.019
	EAD	L_1	1.38	0.578
	HSJA	L_2	0.05	0.015
Robust Neural Network (PGD + HSJA + EAD)	PGD	L_∞	0.02	0.013
	CWL ₂	L_2	1.10	0.090
	EAD	L_1	2.53	1.184
	HSJA	L_2	0.28	0.10

original model: trained on benign data, **PGD Robust Model:** trained using the PGD L_∞ -based attack, **Robust Neural Network:** trained using the PGD, EAD and HSJA attacks

improvement of robustness is favoured against L_2 -based perturbations.

These experimental findings converge to the following answer for our research question: when adversarial training targets on properly selected ensembles of perturbations we achieve increased resilience against all types of adversarial attacks. This is also confirmed by the results in Table 11. Significant improvements are achieved in ψ metric for the adversarially trained model with the combined attacks, compared to the adversarially trained model with the PGD attack

Table 11 ψ adversarial robustness estimates

NN model	AREB	$\psi(x)$	
		MNIST	CIFAR-10
Original model	PGD	0.06	0.07
	CWL ₂	0.85	0.51
	EAD	0.24	1.07
	HSJA	1.17	0.96
PGD robust model	PGD	166.79	41.61
	CWL ₂	22.71	25.24
	EAD	1.06	17.5
	HSJA	51.10	6.25
Robust neural network (PGD + HSJA + EAD)	PGD	130.58	6.33
	CWL ₂	96.92	49.45
	EAD	22.18	108.9
	HSJA	254.16	19.43

Table 12 Loss sensitivity

NN model	AREB	MNIST	CIFAR-10
		Original model	PGD
Original model	CWL ₂	19.56	38.92
	EAD	20.87	32.29
	HSJA	32.83	52.43
	PGD robust model	PGD	0.19
PGD robust model	CWL ₂	0.29	2.37
	EAD	0.90	2.28
	HSJA	0.22	2.32
	Robust neural network (PGD + HSJA + EAD)	PGD	0.37
Robust neural network (PGD + HSJA + EAD)	CWL ₂	0.15	1.66
	EAD	0.43	1.63
	HSJA	0.17	1.58

only. The values for the ψ metric are estimates computed out of 1000 test samples per case (resource demands for computing ψ are less than those for the CLEVER metric). Finally, the aforementioned observations are substantiated by the results of the loss sensitivity metric. As shown in Table 12, the loss gradients are overly steep on average for the original model when subjected to adversarial samples, something that is corroborated by the high metric values. In contrast, the two robust models result in shallow gradients with our model demonstrating superior results in almost all cases.

5 Process flow and cost for robustness improvement

The answer given in Sect. 4.2 for the posed research question and the insight gained through the AREB benchmark give rise to a process flow for holistically improved models against diverse types of adversarial attacks:

- Step 1: assess the adversarial threats for the dataset of interest

The attack parameters with the highest success rate for minimal perturbations (cf. Sect. 2.3.2) have to be fine tuned, for all attacks of the AREB. When the same attack is applied to diverse datasets, the parameter values resulting in the highest success rates may be very different. Various hyperparameter optimization techniques may be

applied, as well as common search procedures such as random search, grid search and others.

- Step 2: assess the effects of preprocess defences on the dataset of interest

The goal is to find out whether any (combination of) preprocess defence(s) can contribute towards improving the robustness of the neural network with respect to the attacks of the AREB. Label smoothing may be an attractive choice, since it does not affect input data, while exhibiting significant improvements according to our experiments. To fine tune the parameter(s) of the preprocess defence(s) under test, the methodology of Sect. 3 will be followed. It is essential to take into account that potential gains in adversarial robustness may come at the expense of classification accuracy for the original data.

- Step 3: create an adversarially trained model
An ensemble adversarial training scheme is to be used, in order to combine attacks with different characteristics and to eventually integrate the resulting model with the preprocess defence chosen in Step 2.

- Step 4: check the robustness improvement with respect to all attacks in the AREB

The adversarially trained model has to retain an adequate level of classification accuracy for the original data, when compared to a neural network that has not been trained on adversarial perturbations. For the robustness improvement, it is advised to use the three metrics described in Sect. 4.1. Such metrics can provide adequate assurance that the adversarially trained network exhibits satisfactory resilience against all adversarial attacks in the AREB.

- Step 5: repeat steps 3 and 4 until being able to achieve the robustness expectations

If a more robust model is needed, alter the attack methods used in adversarial training, while insisting in combinations that represent all adversarial example characteristics, as discussed in Sect. 2.3.

Our experimental work focused on two image classification datasets. However, when the perturbed inputs are visually indistinguishable from benign images, adversarial attacks are very realistic threats for computer vision applications in the physical world. In other application domains, adversarial attacks can be even more serious threats, since the aforementioned condition may not be applicable. In general, we do not see any technical limitations for applying our process flow for improved adversarial robustness to other application domains as well.

5.1 Cost and computational resource demands

The cost of the process flow for holistic improvement of adversarial robustness depends largely on the know-how and

the available tool support for the machine learning engineer. We relied on the IBM Adversarial Robustness Toolbox,¹ an open-source framework that implements all state-of-the-art adversarial attacks and defences. If such a tool/library is available, we estimate that steps 1 and 2 can take up to a few weeks, even if the engineer is not familiar with the tool. On the other hand, if the engineer has already applied the process flow at least once, then for any new attempt to improve the adversarial robustness of a neural network model, steps 1 and 2 can be completed in a few days at most. The parameters' choice largely affects the interpretation of the results in steps 3 and 4, and for this reason, it is essential before moving to the next step to find out appropriate parameter values for the attacks and defences under test.

The computational cost for holistic improvement of adversarial robustness is affected by the fact that adversarial training is in general more computationally demanding than conventional training of neural networks. The computational resources needed depend on the dataset, whereas there is a correlation between the amount of time invested and the robustness improvement achieved. Indicatively, for the MNIST dataset and the results shown in Table 8, it took 30h to train the PGD-based robust model and 150h to train the combined PGD+HSJA+EAD-based robust model. The experiments took place on a dedicated workstation with a 4-core 3 GHz processor, 12GB RAM and the Nvidia RTX2070 graphics card.

6 Conclusion

We presented an approach for holistic improvement of adversarial robustness of deep neural networks. The starting point is our Adversarial Robustness Evaluation Benchmark (AREB) set of attacks, which includes representative cases of the most effective attacks for all aspects of adversarial robustness.

Any approach with a similar focus is inevitably data-dependent; therefore, it was necessary to apply techniques for fine-tuning the parameters of attacks, in order to find those with the highest success rates for minimal data perturbations. An analogous approach was also followed for assessing the effects of preprocess defences. The whole process ends up with combining the most effective preprocess defence, with a robust model obtained via adversarial training using an ensemble training method.

In overall, for both the MNIST and the CIFAR-10 datasets, our process flow resulted in an holistic improvement of adversarial robustness. The achieved improvement has been confirmed by appropriate adversarial robustness metrics; minor exceptions found could be attributed to the preprocess

¹ <https://github.com/Trusted-AI/adversarial-robustness-toolbox>.

defence used. Adversarial robustness can be further improved through the iterative application of the process steps. To conclude, we believe that, if there are available libraries/tools for applying an adequate range of adversarial attacks and defences, as well as sufficient computational resources, our approach to improving adversarial robustness is worth the effort needed, especially when the model under test is a learning component in a critical application/system.

Future research prospects include, but are not limited, to the following challenges. We intend to enrich the L_p metrics used, through including alternatives for quantifying adversarial perturbations, such as the Wasserstein Distance [54] and various computer vision algorithms [55] (Histogram of Oriented Gradients and Edge Detectors). For the adversarially robust models created, we intend to complete the improvement process with a formal robustness verification technique, such as the one reported in [56]. Finally, it is also interesting to explore the specificities for applying the approach to other application domains, such as in natural language processing, speech recognition, deep reinforcement learning and neural controller design, since research on adversarial robustness in these fields has not been advanced, as it has for the image classification problems.^{2, 3}

Acknowledgements This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 956123.

Funding Open access funding provided by HEAL-Link Greece.

Data availability All the data that were utilized in this current work are publicly available and free to use. Specifically, we rely on two image classification datasets, namely MNIST² and CIFAR-10³.

Declarations

Conflict of interest The authors declare no conflict of interest.

Ethical approval Not applicable.

Informed consent Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

² <http://yann.lecun.com/exdb/mnist>.

³ <https://www.cs.toronto.edu/~kriz/cifar.html>.

Appendix A: Experimental setup

All the experiments were conducted on a computer with a 4-core 3GHz processor, 12 GB RAM and a single Nvidia GeForce RTX 2070 8GB graphics card. We build our Deep Neural Networks (DNNs) with TensorFlow⁴ and Keras.⁵

We chose the two most popular datasets in bibliography for image classification problems, in particular MNIST² and CIFAR-10³, in order to expose image data to adversarial threats and test methods that increase models' robustness. MNIST is consisted of 70.000 grayscale images of size $28 \times 28 \times 1$ (width \times height \times channels), CIFAR-10 is composed of 60.000 RGB images of size $32 \times 32 \times 3$, while the data represent 10 object classes in both cases.

We train two models of different complexity per dataset, one Convolutional Neural Network (CNN) with ~ 200.000 parameters and one DNN based on ResNet [6] state-of-the-art architecture. By that means, we aim to investigate if different architecture and additive Neural Networks' *depth* could be a soothing factor against adversarial examples' success rate, as occasionally suggested in the literature.

The CNN for MNIST consists of 2 blocks of the following structure, Convolutional Layer—Batch Normalization Layer—Max Pooling Layer, followed by a Fully Connected Layer of 256 nodes and an Output Layer of 10 nodes with Softmax as the Output Activation Function. The first and second Convolutional Layers' filters depth is 32 and 64, respectively, with no padding, while we keep Pooling filter size of (2, 2) for both Max Pooling Layers.

The CNN for CIFAR-10 consists of 2 blocks of the following structure, Convolutional Layer—Batch Normalization Layer—Convolutional Layer—Batch Normalization Layer—Max Pooling Layer—Dropout Layer, and a block of the same structure with Average Pooling Layer instead of Max Pooling Layer. Then, there is a Fully Connected Layer of 512 nodes and an Output Layer of 10 nodes with Softmax as the Output Activation Function. The first, second and third Convolutional Layers' filters depth is 32, 64, 128, respectively, with padding, while we keep Pooling filter size of (2, 2) for both Max Pooling Layers and (8, 8) for the Average Pooling Layer. The Dropout Layers have a dropout rate (the percentage of nodes to be set to 0 during each training step) of 20%, 20% and 30%, respectively.

All models are trained at batches of 32 assisted with callback functions that control learning-rate decay and Early Stopping [57]. Additionally, we use Adam [58] as the optimizer and Categorical Cross-Entropy as loss function. We apply Data Augmentation as a means of improving generalisation of our models, using the Keras built-in method and by

⁴ <https://www.tensorflow.org>.

⁵ <https://keras.io>.

keeping fixed the method's parameter values for both models per dataset as shown next:

MNIST:

$rotation_range = 20$, $width_shift_range = 0.1$,
 $height_shift_range = 0.1$, $shear_range = 0.1$,
 $zoom_range = 0.1$, $horizontal_flip = True$

CIFAR-10:

$rotation_range = 30$, $width_shift_range = 0.15$,
 $height_shift_range = 0.15$, $shear_range = 0.1$,
 $zoom_range = 0.1$, $horizontal_flip = True$

The MNIST models are trained for 30 epochs with the CNN obtaining 99, 38% and the ResNet 99, 7% test accuracy, respectively. The CIFAR-10 models are trained for 100 epochs, with the resulting test accuracy be 84, 33% for the CNN model and 89, 42% for the ResNet. For adversarial training, we use the same CNN architectures described above, and we add a Fully Connected Layer of 1024 nodes just before the Output Layer, as suggested in [12].

Adversarial attacks

In Appendix B we present the parameter(s) choice for the adversarial attacks of the AREB set, for both MNIST and CIFAR-10 data.

MNIST

PGD: $eps = 0.3$, $eps_step = 0.01$, $max_iter = 40$
 CWL2: $confidence = 0.05$, $initial_const = 0.1$,
 $binary_search_steps = 20$
 EAD: $confidence = 0.5$, $max_iter = 10$,
 $binary_search_steps = 20$
 HSJA: $max_iter = 10$, $max_eval = 100$

CIFAR-10

PGD: $eps = 0.03$, $eps_step = 0.008$, $max_iter = 7$
 CWL2: $confidence = 0.2$, $initial_const = 0.01$,
 $binary_search_steps = 20$
 EAD: $confidence = 0.4$, $max_iter = 10$,
 $binary_search_steps = 20$
 HSJA: $max_iter = 10$, $max_eval = 100$

Preprocess defences

In Appendix C we present the hyperparameter values for the preprocess defences deployed in MNIST (see Table 12) and CIFAR-10 (see Table 13). This particular selection is applied for both models (CNN, ResNet) on these two datasets.

References

1. Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., Roli, F.: Evasion attacks against machine learning at test time. Lecture Notes in Computer Science, 387–402 (2013) https://doi.org/10.1007/978-3-642-40994-3_25
2. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R.: Intriguing properties of neural networks. In: Bengio, Y., LeCun, Y. (eds.) 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14–16, 2014, Conference Track Proceedings (2014). [arXiv:1312.6199](https://arxiv.org/abs/1312.6199)
3. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings (2015). [arXiv:1412.6572](https://arxiv.org/abs/1412.6572)
4. Papernot, N., McDaniel, P., Goodfellow, I.: Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. [arXiv:1605.07277](https://arxiv.org/abs/1605.07277) (2016)
5. Biggio, B., Nelson, B., Laskov, P.: Support vector machines under adversarial label noise. In: Hsu, C.-N., Lee, W.S. (eds.) Proceedings of the Asian Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 20, pp. 97–112. PMLR, South Garden Hotels and Resorts, Taoyuan, Taiwan (2011). <http://proceedings.mlr.press/v20/biggio11.html>
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016). <https://doi.org/10.1109/CVPR.2016.90>
7. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1. NIPS'15, pp. 91–99. MIT Press, Cambridge, MA, USA (2015)
8. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: Unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779–788 (2016)
9. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. In: NAACL (2019)
10. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019)
11. Ren, H., Huang, T.: Adversarial example attacks in the physical world. In: Machine Learning for Cyber Security: Third Inter-

Table 13 Preprocess defences hyperparameter choice on MNIST

Preprocess defence	Hyperparameter	Value
Label smoothing	k	0.9
Feature squeezing	Bit depth	1
JPEG compression	Quality	6%
Spatial smoothing	Window size	4
Total variance minimization	λ	0.2

Table 14 Preprocess defences hyperparameter choice on CIFAR-10

Preprocess defence	Hyperparameter	Value
Label smoothing	k	0.9
Feature squeezing	Bit depth	5
JPEG compression	Quality	85%
Spatial smoothing	Window size	2
Total variance minimization	λ	0.09

- national Conference, ML4CS 2020, Guangzhou, China, October 8–10, 2020, Proceedings, Part II, pp. 572–582. Springer, Berlin (2020). https://doi.org/10.1007/978-3-030-62460-6_51
12. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (2018). <https://openreview.net/forum?id=rJzIBfZAb>
 13. Croce, F., Hein, M.: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: Proceedings of the 37th International Conference on Machine Learning. ICML'20 (2020)
 14. Chen, P.-Y., Sharma, Y., Zhang, H., Yi, J., Hsieh, C.-J.: Ead: Elastic-net attacks to deep neural networks via adversarial examples. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
 15. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. 2016 IEEE European Symposium on Security and Privacy (EuroS&P), 372–387 (2016)
 16. Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 86–94 (2017)
 17. Moosavi-Dezfooli, S.-M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2574–2582 (2016). <https://doi.org/10.1109/CVPR.2016.282>
 18. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 39–57. IEEE Computer Society, Los Alamitos, CA, USA (2017). <https://doi.org/10.1109/SP.2017.49>
 19. Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., Hsieh, C.-J.: Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. AISEC '17, pp. 15–26. Association for Computing Machinery, New York, NY, USA (2017). <https://doi.org/10.1145/3128572.3140448>
 20. Brendel, W., Rauber, J., Bethge, M.: Decision-based adversarial attacks: reliable attacks against black-box machine learning models. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30–May 3, 2018, Conference Track Proceedings (2018). <https://openreview.net/forum?id=SyZiOGWCZ>
 21. Chen, J., Jordan, M.I., Wainwright, M.J.: Hopskipjumpattack: A query-efficient decision-based attack. In: 2020 IEEE Symposium on Security and Privacy (SP), pp. 1277–1294. IEEE Computer Society, Los Alamitos, CA, USA (2020). <https://doi.org/10.1109/SP40000.2020.00045>
 22. Cheng, M., Singh, S., Chen, P.H., Chen, P.-Y., Liu, S., Hsieh, C.-J.: Sign-opt: a query-efficient hard-label adversarial attack. In: International Conference on Learning Representations (2020). <https://openreview.net/forum?id=SkITQCNTvS>
 23. Andriushchenko, M., Croce, F., Flammarion, N., Hein, M.: Square attack: a query-efficient black-box adversarial attack via random search. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) Computer Vision - ECCV 2020, pp. 484–501. Springer, Cham (2020)
 24. Guo, C., Gardner, J.R., You, Y., Wilson, A.G., Weinberger, K.Q.: Simple black-box adversarial attacks. CoRR [arXiv:1905.07121](https://arxiv.org/abs/1905.07121) (2019)
 25. Serban, A., Poll, E., Visser, J.: Adversarial examples on object recognition: A comprehensive survey. ACM Comput. Surv. **53**(3) (2020) <https://doi.org/10.1145/3398394>
 26. Kos, J., Fischer, I., Song, D.: Adversarial examples for generative models. In: 2018 IEEE Security and Privacy Workshops (SPW), pp. 36–42 (2018). <https://doi.org/10.1109/SPW.2018.00014>
 27. Xiao, C., Li, B., Zhu, J.-Y., He, W., Liu, M., Song, D.: Generating adversarial examples with adversarial networks. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence. IJCAI'18, pp. 3905–3911 (2018)
 28. Hashemi, A.S., Mozaffari, S.: Secure deep neural networks using adversarial image generation and training with noise-gan. Comput. Secur. **86**, 372–387 (2019). <https://doi.org/10.1016/j.cose.2019.06.012>
 29. Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., Song, D.X.: Robust physical-world attacks on deep learning visual classification. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 1625–1634 (2018)
 30. Song, D., Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Tramèr, F., Prakash, A., Kohno, T.: Physical adversarial examples for object detectors. In: 12th USENIX Workshop on Offensive Technologies (WOOT 18). USENIX Association, Baltimore, MD (2018). <https://www.usenix.org/conference/woot18/presentation/eykholt>
 31. Sitawarin, C., Bhagoji, A.N., Mosenia, A., Chiang, M., Mittal, P.: Darts: deceiving autonomous cars with toxic signs. [arXiv:1802.06430](https://arxiv.org/abs/1802.06430) (2018)
 32. Zhao, Y., Zhu, H., Liang, R., Shen, Q., Zhang, S., Chen, K.: Seeing isn't believing: Towards more robust adversarial attack against real world object detectors. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. CCS '19, pp. 1989–2004. Association for Computing Machinery, New York, NY, USA (2019). <https://doi.org/10.1145/3319535.3354259>
 33. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. In: International Conference on Learning Representations (2019). <https://openreview.net/forum?id=HJz6tiCqYm>
 34. Hashemi, A., Mozaffari, S.: Cnn adversarial attack mitigation using perturbed samples training. Multimedia Tools Appl. **80**, 22077–22095 (2021). <https://doi.org/10.1007/s11042-020-10379-6>
 35. Croce, F., Andriushchenko, M., Sehwag, V., DeBenedetti, E., Flammarion, N., Chiang, M., Mittal, P., Hein, M.: Robustbench: a standardized adversarial robustness benchmark. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2) (2021). <https://openreview.net/forum?id=SSKZPJc7B>
 36. Chen, J., Gu, Q.: Rays: A ray searching method for hard-label adversarial attack. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD '20, pp. 1739–1747. Association for Computing Machinery, New York, NY, USA (2020). <https://doi.org/10.1145/3394486.3403225>
 37. Ling, X., Ji, S., Zou, J., Wang, J., Wu, C., Li, B., Wang, T.: Deepsec: a uniform platform for security analysis of deep learning model. In: 2019 IEEE Symposium on Security and Privacy (SP), pp. 673–690 (2019)
 38. Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., Madry, A.: Robustness may be at odds with accuracy. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6–9, 2019 (2019). <https://openreview.net/forum?id=SyxAb30cY7>
 39. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009). <https://doi.org/10.1109/CVPR.2009.5206848>
 40. Su, J., Vargas, D.V., Sakurai, K.: One pixel attack for fooling deep neural networks. IEEE Trans. Evol. Comput. **23**(5), 828–841 (2019). <https://doi.org/10.1109/TEVC.2019.2890858>
 41. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. J. R. Stat. Soc. Ser. B (Statistical Methodology) **67**(2), 301–320 (2005)

42. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2818–2826 (2016). <https://doi.org/10.1109/CVPR.2016.308>
43. Xu, W., Evans, D., Qi, Y.: Feature squeezing: detecting adversarial examples in deep neural networks. In: Proceedings 2018 Network and Distributed System Security Symposium (2018) <https://doi.org/10.14722/ndss.2018.23198>
44. Dziugaite, G.K., Ghahramani, Z., Roy, D.M.: A study of the effect of jpg compression on adversarial images. [arXiv:1608.00853](https://arxiv.org/abs/1608.00853) (2016)
45. Das, N., Shanbhogue, M., Chen, S.-T., Hohman, F., Chen, L., Kounavis, M.E., Chau, D.H.: Keeping the bad guys out: protecting and vaccinating deep learning with jpeg compression. [arXiv:1705.02900](https://arxiv.org/abs/1705.02900) (2017)
46. Guo, C., Rana, M., Cisse, M., Maaten, L.: Countering adversarial images using input transformations. In: International Conference on Learning Representations (2018). <https://openreview.net/forum?id=SyJ7CIWCb>
47. Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I.J., Boneh, D., McDaniel, P.D.: Ensemble adversarial training: Attacks and defenses. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30–May 3, 2018, Conference Track Proceedings (2018). <https://openreview.net/forum?id=rkZvSe-RZ>
48. Tramer, F., Boneh, D.: Adversarial training and robustness for multiple perturbations. In: Advances in Neural Information Processing Systems, vol. 32 (2019). <https://proceedings.neurips.cc/paper/2019/file/5d4ae76f053f8f2516ad12961ef7fe97-Paper.pdf>
49. Maini, P., Wong, E., Kolter, J.Z.: Adversarial robustness against the union of multiple perturbation models. In: Proceedings of the 37th International Conference on Machine Learning. ICML'20 (2020)
50. Weng, T., Zhang, H., Chen, P., Yi, J., Su, D., Gao, Y., Hsieh, C., Daniel, L.: Evaluating the robustness of neural networks: An extreme value theory approach. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30–May 3, 2018, Conference Track Proceedings (2018). <https://openreview.net/forum?id=BkUHIMZ0b>
51. Yu, F., Qin, Z., Liu, C., Zhao, L., Wang, Y., Chen, X.: Interpreting and evaluating neural network robustness. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. IJCAI'19, pp. 4199–4205 (2019)
52. Arpit, D., Jastrzundefinedbski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M.S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., Lacoste-Julien, S.: A closer look at memorization in deep networks. In: Proceedings of the 34th International Conference on Machine Learning—Volume 70. ICML'17, pp. 233–242 (2017)
53. Gao, R., Liu, F., Zhang, J., Han, B., Liu, T., Niu, G., Sugiyama, M.: Maximum mean discrepancy test is aware of adversarial attacks. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18–24 July 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 139, pp. 3564–3575 (2021). <http://proceedings.mlr.press/v139/gao21b.html>
54. Sinha, A., Namkoong, H., Duchi, J.: Certifiable distributional robustness with principled adversarial training. In: International Conference on Learning Representations (2018). <https://openreview.net/forum?id=Hk6kPgZA->
55. Jang, U., Wu, X., Jha, S.: Objective metrics and gradient descent algorithms for adversarial examples in machine learning. In: Proceedings of the 33rd Annual Computer Security Applications Conference. ACSAC 2017, pp. 262–277. Association for Computing Machinery, New York, NY, USA (2017). <https://doi.org/10.1145/3134600.3134635>
56. Ruan, W., Wu, M., Sun, Y., Huang, X., Kroening, D., Kwiatkowska, M.: Global robustness evaluation of deep neural networks with provable guarantees for the Hamming distance. In: Kraus, S. (ed.) International Joint Conference on Artificial Intelligence (IJCAI), pp. 5944–5952 (2019)
57. Caruana, R., Lawrence, S., Giles, C.L.: Overfitting in neural nets: backpropagation, conjugate gradient, and early stopping. In: NIPS (2000)
58. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings (2015). [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Charis Eleftheriadis is a PhD candidate at the School of Informatics of the Aristotle University of Thessaloniki (AUTH), Greece. He holds a Diploma in Industrial Engineering & Management from the Democritus University of Thrace and a M.Eng in Computational Intelligence and Applications from AUTH. His research interests lie in the areas of Adversarial Machine Learning, AI safety and security, and formal methods for machine learning.

Andreas Symeonidis is Associate Professor at the School of Electrical and Computer Engineering at the Aristotle University of Thessaloniki and the Chief Research Officer at Cyclopt.com. He received his diploma and PhD at the School of Electrical and Computer Engineering at the Aristotle University of Thessaloniki in 1999 and 2004 respectively and concluded his postdoctoral research on the evaluation of agent efficiency (2005). His research interests lie in the areas of software engineering processes, model-driven engineering, software quality and software analytics, middleware robotics and knowledge extraction from big data repositories. Dr. Symeonidis' work has been published in over 200 papers, book chapters, and conference publications. He is currently coordinating more than 10 contract R&D projects, while serving occasionally as a R&D project evaluator and reviewer for the European Commission and the Greek state.

Panagiotis Katsaros is Associate Professor at the School of Informatics of the Aristotle University of Thessaloniki (AUTH), Greece. He holds a Bachelor in Mathematics from AUTH, a Master of Science on Software Engineering from Aston University in Birmingham and a PhD in Computer Science from AUTH. Prof. Katsaros has published over 100 research papers in international journals and conference proceedings on Software Engineering. His specific interests, including those for software engineering for the Internet of Things, refer to the formal verification of software/systems, the model-based design, the analysis of dependability and security and the simulation-based performance analysis and optimization. Prof. Katsaros is coordinator (or participates in) national and European R&D projects focusing on engineering of software for Internet of Things systems, space systems and more recently autonomous systems. Regular updates on Prof. Katsaros recent research achievements can be accessed online at: <https://depend.csd.auth.gr>