




Research Article

Challenging the Adversarial Robustness of DNNs Based on Error-Correcting Output Codes

Bowen Zhang ¹, Benedetta Tondi,² Xixiang Lv ¹ and Mauro Barni ²

¹School of Cyber Engineering, Xidian University, Xi'an 710126, China

²Department of Information Engineering and Mathematics, University of Siena, Siena 53100, Italy

Correspondence should be addressed to Xixiang Lv; xxlv@mail.xidian.edu.cn

Received 21 August 2020; Revised 21 September 2020; Accepted 7 October 2020; Published 16 November 2020

Academic Editor: Zhihua Xia

Copyright © 2020 Bowen Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The existence of adversarial examples and the easiness with which they can be generated raise several security concerns with regard to deep learning systems, pushing researchers to develop suitable defence mechanisms. The use of networks adopting error-correcting output codes (ECOC) has recently been proposed to counter the creation of adversarial examples in a white-box setting. In this paper, we carry out an in-depth investigation of the adversarial robustness achieved by the ECOC approach. We do so by proposing a new adversarial attack specifically designed for multilabel classification architectures, like the ECOC-based one, and by applying two existing attacks. In contrast to previous findings, our analysis reveals that ECOC-based networks can be attacked quite easily by introducing a small adversarial perturbation. Moreover, the adversarial examples can be generated in such a way to achieve high probabilities for the predicted target class, hence making it difficult to use the prediction confidence to detect them. Our findings are proven by means of experimental results obtained on MNIST, CIFAR-10, and GTSRB classification tasks.

1. Introduction

Deep neural networks can solve complicated computer vision tasks with unprecedented high accuracies. However, they have been shown to be vulnerable to *adversarial examples*, namely, properly crafted inputs introducing small (often imperceptible) perturbations, inducing a classification error [1–3]. The possibility of crafting both nontargeted and targeted attacks has been demonstrated, the goal of the former being to induce any kind of classification error [4, 5], while the latter aims at making the network decide for a target class chosen a priori [1, 6]. It goes without saying that, in general, targeted attacks are more difficult to build.

As a reaction to the threats posed by adversarial examples, many defence mechanisms have been proposed to increase the adversarial robustness of deep neural networks [7–12]. However, in a white-box setting wherein the attacker has a full knowledge of the attacked network, including full knowledge of the defence mechanism, more powerful attacks can be developed, thus tipping again the scale in favour of the attacker [4, 13].

In this race of arms, a novel defence strategy based on *error-correcting output coding* (ECOC) [14] has been proposed recently in [15], to counter adversarial attacks in a white-box setting. More specifically, given a general multiclass classification problem, error-correcting output codes are used to encode the various classes and represent the network's outputs. To explain how, let us refer to the output of the last layer of the network, prior to the final activation layer, as logit values or simply logits. In general, the final activation layer consists of the application of an activation function, which maps the logits into a prescribed range, and a normalization layer, which maps the output of the activation functions into a probability vector, associating a probability value to each class. In the common case of one-hot-encoding, a softmax layer is used, in which case these two steps are performed simultaneously. During training, the network learns to output a large logit value for the true class and small values for all the others. With the ECOC approach, instead, the network is trained in such a way to produce normalized logit values that correlate well with the codeword used to encode the class the input sample belongs

to. In general, ECOC codewords have many nonzero values, thus marking a significant difference with respect to the one-hot-encoding case.

The rationale behind the use of the ECOC architecture to counter the construction of adversarial examples [15] is that while with classifiers based on standard one-hot-encoding the attacker can induce an error by modifying one single logit (reducing the one associated to the ground-truth class or increasing the one associated to the target class), the final decision of the ECOC classifier depends on multiple logits in a complicated manner, and hence it is supposedly more difficult to attack (especially when longer codewords are used).

In [15], the authors considered nontargeted attacks in their experiments and showed with the popular white-box C&W attack that the attack success rate on CIFAR-10 [16] passes from 100%, for one-hot-encoding, to 29%, for an ECOC-based classifier.

Another alleged advantage of the ECOC architecture proposed in [15] is linked to the way the probabilities associated with each class are computed. Rather than using a softmax function as commonly done with one-hot-encoding, first the correlation between the activated outputs and the codeword is computed, and then a linear normalization procedure is applied (see equation (2) in the following). In this way, the probability assigned to the class chosen by the classifier grows more smoothly, and samples close to the decision region boundary (like adversarial examples are likely to be) are classified with a low confidence. Results presented in [15], in fact, show that the ECOC model tends to provide sharp results for clean images, while it is often uncertain about the (incorrect) prediction made on adversarial examples. This behavior could be exploited to, at least, distinguish between adversarial examples and benign inputs.

The goal of this paper is to further verify if and to which extent the use of error correction codes to encode the output of deep neural networks allows to increase the robustness against targeted adversarial examples. We do so by introducing a new white-box attack, inspired to C&W attack, explicitly thought to work not only against ECOC but also other multilabel classifiers. In fact, the original C&W is naturally designed to deceive networks adopting the one-hot-encoding strategy, and it loses some of its advantages when used against ECOC systems. We stress that, in contrast to previous works (see, for instance, [15] and Section 10 in [17]), we aim at developing a targeted attack, which is a more difficult task than crafting nontargeted adversarial examples. This is a reasonable choice for at least two reasons. First, targeted attacks are more flexible than nontargeted ones since they can be used in a wider variety of applications, wherein the ultimate goal of the attack may vary considerably. Secondly, being able to attack a defence under most stringent attacking constraints illustrates better the weakness of the defence itself.

We ran extensive experiments to evaluate the ability of ECOC-based classifiers to resist the new attack and compared the results we got with those obtained by applying a fine-tuned version of C&W attack and the LOTS attack

introduced in [18]. The experiments were carried out by considering three different classification tasks, namely, traffic sign classification (GTSRB) [19], CIFAR-10 classification [16], and MNIST [20]. As a result, we found that the ECOC classifiers can be successfully attacked with a high success rate. In particular, the new attack outperforms the other two especially when long codewords are used by ECOC. We also verified that, by increasing the confidence of the attack, adversarial examples can achieve high probabilities for the predicted target class, similar to those of benign samples, hence making it difficult to use the prediction confidence to detect adversarial samples. Overall, our analysis reveals that the security gain achieved by the ECOC scheme is a minor one, thus calling for more powerful defences.

The rest of this paper is organised as follows: we first briefly review the ECOC scheme presented in [15], and then we describe the proposed attack. The setup considered for the experiments, and the results we got are reported and discussed in Section 4. Eventually, we review the related work at the end of the paper.

2. ECOC-Based Classification

Let us first introduce the notation for a general multiclass CNN. Let x be the input of the network and k the class label, $k = 1, 2, \dots, M$, where M denotes the number of classes. Let $f(x)$ indicates the decision function of the network. We denote by $\mathbf{z} = (z_1, z_2, \dots)$, the vector with the logit values, that is, the network values before the final activations and the mapping to class probabilities. For one-hot-encoding schemes, \mathbf{z} has length M , and the logits are directly mapped into probability values through the softmax function ψ as follows:

$$p_\psi(k) = \psi_k(\mathbf{z}) = \frac{\exp(z_k)}{\sum_{i=1}^M \exp(z_i)}, \quad (1)$$

for $k = 1, \dots, M$. Then, the final prediction is made by letting $f(x) = \arg \max_k p_\psi(k)$.

The error-correction-output-coding (ECOC) scheme proposed in [15] assigns a codeword \mathbf{C}_k of length N ($N \geq M$) to every output class ($k = 1, \dots, M$). \mathbf{C} denotes the $M \times N$ codeword matrix. Each element of \mathbf{C} can take values in $\{-1, 1\}$. In this way, the length of the logit vector \mathbf{z} is N . The logits are first mapped into the $[-1, 1]$ range by means of an activation function $\sigma(\cdot)$ (e.g., the tanh function that $\sigma(x) = (e^x - e^{-x}) / (e^x + e^{-x})$). Then, the probability of class k is computed by looking at the correlation with \mathbf{C}_k , according to the following formula:

$$p_\sigma(k) = \frac{\max(\sigma(\mathbf{z}) \cdot \mathbf{C}_k, 0)}{\sum_{i=1}^M \max(\sigma(\mathbf{z}) \cdot \mathbf{C}_i, 0)}, \quad (2)$$

where \cdot denotes the inner product and $\sigma(\cdot)$ is a sigmoid activation function applied element-wise to the logits. Since \mathbf{C}_{ij} s take values in $\{-1, 1\}$, the max is necessary to avoid negative probabilities. According to [15], the common softmax rule (equation (1)) is able to express uncertainty between two classes only when the logits are roughly equal

(i.e., $z_1 \approx z_2$ and the two probabilities are close $p_\psi(i) \approx p_\psi(j)$). In a two dimensional case, this corresponds to a very narrow stripe, approximate to a line, across the boundary of the decision region, while in high dimensional spaces, the region $z_i \approx z_j$ approximates a hyperplan, an $M - 1$ dimensional subspace of \mathbb{R}^M with negligible volume, and hence the classifier outputs high probabilities almost everywhere. This makes it very easy for the attacker to find an adversarial input that is predicted (wrongly) with high confidence. With ECOC (equation (2)), instead, it is sufficient that two approximate correlations express low uncertainty ($\sigma(\mathbf{z}) \cdot \mathbf{C}_i \approx \sigma(\mathbf{z}) \cdot \mathbf{C}_j$), and then a non-trivial volume is allocated to low-confidence region in the logit space, thus limiting the freedom of the attacker to craft high-confidence adversarial examples. An overall sketch of the ECOC scheme is depicted in Figure 1. The logits \mathbf{z} are first mapped into correlation values, $\rho := \sigma(\mathbf{z}) \cdot \mathbf{C}$ (mapping step 1), and then the vector with the correlations is normalized so to form a probability distribution (mapping step 2) via the normalization function in (2). The model's final predicted label is $\arg \max_k p_\sigma(k)$. Equation (2) is a generalization of the standard softmax activation in equation (1) and reduces to it for the case of one-hot-encoding, that is, when $\mathbf{C} = \mathbf{I}_M$, with $N = M$, and where \mathbf{I}_M is the identity $M \times M$ matrix.

The purpose of the ECOC architecture is to design a classifier which is robust to changes of multiple logits and then, expectedly, more difficult to attack (with standard one-hot-encoding the adversary can succeed by altering a single logit). For the scheme to be effective, codewords characterised by a large minimum Hamming distance must be chosen. For simplicity, in [15], the ECOC classifier is built by using Hadamard codes taking values in $\{-1, 1\}$ (when \mathbf{C} is a Hadamard matrix, the Hamming distance for large M approaches the limit value $N/2$). An advantage with this choice is that, since \mathbf{C} is orthogonal, whenever the network outputs a codeword exactly (that is when $\sigma(\mathbf{z}) = \mathbf{C}_k$), then $p_\sigma(k) = 1$. The tanh function is selected as the activation function $\sigma(\cdot)$.

The authors also found that, rather than considering a single network with N outputs, a classifier consisting of an ensemble of several smaller networks, each one outputting a few codeword elements, permits to achieve a larger robustness against attacks. By training separate networks, in fact, the correlation between errors affecting different bits of the codewords is reduced, thus forcing the attacker to attack all the bits independently. In the scheme in Figure 1, every network outputs one codeword bit only, resulting in N ensemble branches.

3. Attacking ECOC

We start by considering the basic C&W attack introduced in [6]. We notice that some of the good properties of C&W do not hold longer when the attack is applied against the ECOC scheme since it has been originally designed to work against networks adopting the one-hot-encoding strategy. Then, we propose a new more effective attack, which is specially tailored to multilabel structures like ECOC.

In general, constructing an adversarial example corresponds to finding a small perturbation δ (under some

distance metric) that once added to image x will change its classification. Such a problem is usually formalised as

$$\begin{aligned} \min D(x, x + \delta), \\ \text{s.t. } f(x + \delta) = t, \end{aligned} \quad (3)$$

where D is some distance metric (e.g. the L_2 metric) and t is a chosen target class. As this problem is difficult to solve, C&W attack aims at solving its Lagrangian approximation defined as

$$\min \|\delta\|_2 + \lambda \cdot \max \left(\max_{i \neq t} (z_i(x + \delta)) - z_t(x + \delta), c \right), \quad (4)$$

where the second term is any function such that $f(x + \delta) = t$ if and only if this term $\leq c$. $\|\cdot\|_2$ denotes the L_2 -norm, λ and c are constant parameters ruling, respectively, the tradeoff between the two terms of the optimization problem and the confidence margin of the attack (In [6], $\delta = 1/2(\tanh(w) + 1) - x$, and the minimization is carried out over w to have box constraints on δ when optimizing equation (4) with a common optimizer like Adam.). Equation (4) is designed for the common one-hot encoding case. In fact, it is easy to see that for ECOC the motivation of such a design does not hold anymore and ensure that the second term is less than c and does not guarantee that $f(x + \delta) = t$. Therefore, the C&W attack must be adjusted to fit the ECOC framework. By noting that, in ECOC, correlations are proportional to probabilities (instead of the logits as with one-hot encoding), and C&W shall be modified as

$$\underset{\delta}{\text{minimize}} \|\delta\|_2 + \lambda \cdot \max \left(\max_{i \neq t} (\rho_i(x + \delta)) - \rho_t(x + \delta), c \right), \quad (5)$$

where $\rho_i(x + \delta) = \sigma(\mathbf{z}(x + \delta)) \cdot \mathbf{C}_i$.

A key advantage of C&W attack against one-hot-encoding networks is that it works directly at the logits level. In fact, logits are more sensitive to modifications of the input than the probability distribution obtained after the softmax activation (most adversarial attacks work directly on the probability values obtained after the softmax, which makes them less effective than C&W and prone to gradient-vanishing problems).

When C&W attack is applied against ECOC (by means of (5)), it does not work at the output logit level, but after that, the correlations are computed (mapping step 1) since this is the layer that precedes the application of the softmax-like function. The correlations between the activations of the logits and the codewords will likely have a reduced sensitivity to input modifications, and this may decrease the effectiveness of the attack. We also found that during the attack, it is possible to change only one bit of the output while the others are almost unchanged. This can be explained by observing that ECOC trains each output bit separately, so that each bit can be treated as an individual label. In this way, the correlation between the output bits is significantly reduced compared to classifiers adopting the one-hot encoding approach. We exploit this fact to design our attack in such a way as to make it modify a single bit at a

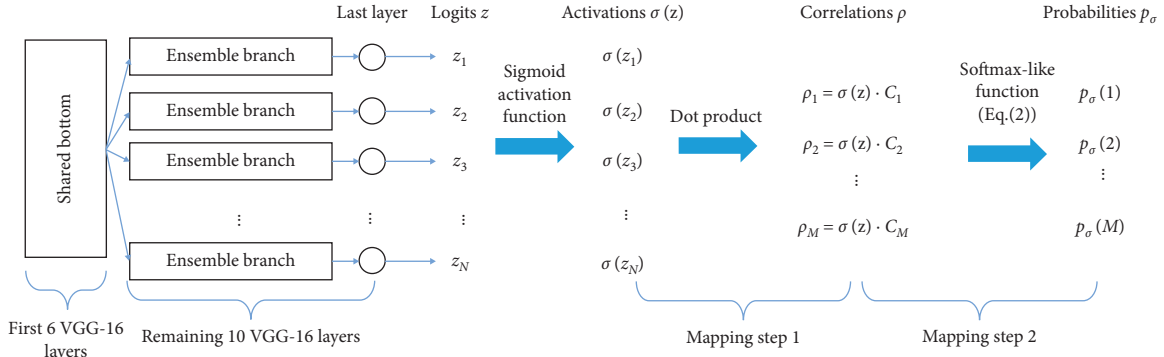


FIGURE 1: Block diagram of ECOC architecture.

time and iteratively repeat this process to eventually change multiple bits.

With the above ideas in mind, the new attack is formulated as follows:

$$\underset{\delta}{\text{minimize}} \|\delta\|_2 - \lambda \cdot \min_i (2t_i \cdot z_i(x + \delta), c), \quad (6)$$

where $(t_1, t_2, \dots, t_N) = \mathbf{C}_t$ is the desired target codeword ($t_i \in \{-1, 1\}$), λ is a parameter controlling the tradeoff between the two terms of the objective function, and c is a constant parameter used to set a confidence threshold for the attack. Specifically, the attack seeks to minimize (6) until the product between t_i and z_i reaches this threshold; thus, a higher c will result in adversarial examples exhibiting a higher correlation with the target codeword, that is, adversarial examples that are (wrongly) classified with a higher confidence.

The choice of λ also plays an important role in the attack, given that a very small λ would lead to a vanishing perturbation. On the contrary, using larger λ results in a more effective attack at the cost of a larger perturbation. To optimize the value of λ , we use a binary search similar to the one used in [6] to determine the optimum value of λ in C&W attack. By doing so, the parameters of the proposed attack have the same meaning of those in C&W attack; thus, the two methods can be compared on a fair basis under the same parameter setting. An overall description of our attack is given in Algorithm 1, whose goal is to find a valid adversarial example, with the desired confidence level c and with the smallest perturbation. As a result of the optimization in Algorithm 1, all logit values z_i of the resulting adversarial image will tend to be highly correlated with t_i .

It is worth observing that, even if we designed the new attack explicitly targeting the ECOC classifier, the algorithm in (6) is generally applicable to any multilabel classification network since it manipulates the output bits of the network, regardless of the adopted coding strategy. This point can be evidenced by considering two limit cases of ECOC. In the first case, we avoid using error correction to encode the output classes. This is equivalent to multilabel classification problems with N labels [21, 22], and the proposed attack can still be applied. In the second case, we may consider one-hot-encoding as a particular way of encoding the output class.

This perspective, also been considered in [15], would degrade the ECOC system to a common network that uses one-hot-encoding and softmax to solve a multiclass classification task. Since our attack does not involve the decoding part of the network, it can still be applied to such networks.

4. Experiments

4.1. Methodology. In [15], the authors tested the robustness of the ECOC architecture for various combinations of codeword matrices \mathbf{C} , activation functions $\sigma(\cdot)$, and network structures. In particular, they considered the MNIST [20] and CIFAR-10 [16] classification tasks ($M=10$ in both cases). In the end, the best performing system was obtained by considering a Hadamard code with $N=16$ and the tanh activation function. An ensemble of 4 ($N/4$) networks each one outputting 4 bits was considered. The authors argue that using a large number of ensembles increases the performance of the system against attacks (by decreasing the dependency among output bits). Then, in our experiments, we used N ensembles, with only one output bit each. The authors also indicate that the robustness of ECOC scheme can be improved by using longer codewords. Then, in our experiments, in addition to MNIST and CIFAR-10 already considered in [15], we also considered traffic sign classification (GTSRB dataset) [19], to test the robustness of ECOC on a larger number of classes and with codewords of a larger size, which potentially means higher robustness. To be specific, for traffic sign classification, we set $M=32$, by selecting the classes with more examples among the total number of 44 classes in GTSRB, and chose a Hadamard code with $N=M=32$, which is twice the size of the code used for MNIST and CIFAR-10. A diagram of the ECOC scheme with the N ensemble structure is shown in Figure 1. We used a standard VGG-16 network [23] as the base block of our implementation. Following the ECOC design scheme, the first 6 layers form the so-called *shared bottom* part, that is, the layers shared by all the networks of the ensemble. Then, the remaining 10 layers (the last 8 convolutional layers and the 2 fully connected layers) are trained separately for each ensemble branch.

For each task, we first trained one M -class classification network, and then we fine-tuned the weights to get the N

ensemble networks. The error rates of the trained models on clean images are equal to 2.14% for MNIST, 13.9% for CIFAR-10, and 1.28% for traffic sign (GTSRB) classification.

In addition to the extended C&W attack described in Section 3, we also considered a new attack named layerwise origin-target synthesis (LOTS) introduced in [18]. In a few words, LOTS aims at modifying the deep representation at a chosen attack layer, by minimizing the Euclidian distance between the deep features of the to-be-attacked input and a target deep representation chosen by the attacker. In our tests, we applied LOTS to the logits level, and we obtained the target deep representation (logits) by randomly choosing 50 images belonging to the target class.

4.2. Results. We attacked 300 images randomly chosen from the test set of each task. For each attack, we carried out a targeted attack with the target class chosen at random among the remaining $M - 1$ classes (i.e., all the M classes except the original class of the unperturbed image). The label t of the target class was used to run the C&W attack in equation (5) and LOTS, while the codeword C_t associated to t is considered in (6) for the new attack. We use the attack success rate (ASR) to measure the effectiveness of the attack, i.e., the percentage of generated adversarial examples that are assigned to the target class, and the peak signal-to-noise ratio (PSNR) to measure the distortion introduced by the attack, which is defined as $\text{PSNR} = 20 \cdot \log_{10} (255 \cdot \sqrt{N} / \|\delta\|_2)$, where $\|\delta\|_2$ is the L_2 -norm of the perturbation and N is the size of the image.

As the parameters of the new attack have the same meaning as those of C&W attack, we first compare the C&W and the new attack with several settings of the input parameters. The results we got are shown in Tables 1–3, for GTSRB, CIFAR-10, and MNIST, respectively. In all the cases, c was set to 0. The results obtained by using the C&W attack against the standard one-hot-encoding VGG-16 network with M classes are also reported in the last column. By looking at the different rows, we can first see that when the strength of the attack is increased, e.g., by using a larger number of iterations or a larger number of steps during the binary search, the ASR of both attacks increases, at the price of a slightly larger distortion. For instance, for CIFAR-10, the ASR of the proposed attack increases from 69.3% to 98.6%, with a decrease in the PSNR of less than 1 dB, and the ASR of the C&W attack increases from 53.6% to 92.6% with an extra distortion of 3 dB. Then, by comparing different columns, we see a clear advantage of the proposed attack over C&W attack since the former achieves a higher ASR for the same parameter settings.

By comparing the different tables, we see that the advantage of the new attack is more evident with GTSRB than with CIFAR-10. The use of longer codewords in GTSRB, in fact, makes it harder to attack this classifier; however, the new attack can still achieve an ASR = 93.3% with a PSNR equal to 39 dB.

For MNIST dataset, the ASR is lower compared to the CIFAR-10 and GTSRB. This result agrees with the results reported in [15]. One possible explanation of this fact is advanced in [10] where the peculiarities of the MNIST

dataset are highlighted and used to argue that high robustness can be easily reached on MNIST.

The comparison with LOTS must be carried on a different ground since such an attack is designed in a different way, and the only parameter shared with the new attack is the maximum number of iterations allowed in the gradient descent. For this reason, we applied LOTS by allowing a maximum number of iterations equal to 2000, which is the same number we have used for the other two attacks. We have verified experimentally that LOTS converges within 1000 iterations 92% of the times (The convergence is determined by checking whether the new loss value is close enough to the average loss value of the last 10 iterations.), thus validating the adequacy of our choice. Then, we measured the ASR for a given maximum PSNR, thus allowing us to plot the ASR as a function of PSNR. The results we got are shown in Figure 2. Upon inspection of the figure, we observe a behavior similar to Tables 1–3. The proposed attack greatly outperforms LOTS and C&W on GTSRB when longer codewords are used. The ASR of the new attack, in fact, achieves nearly 100% for smaller PSNR's, while LOTS and C&W stop at 42% and 42.3%, respectively. For the other two datasets, the gap between the different attacks is smaller than in the GTSRB case. Specifically, the proposed attack and LOTS perform almost the same on CIFAR-10, while LOTS provides slightly better results on MNIST. This observation can also be verified in Figure 3, where we show some images that are successfully attacked by all the attacks. We can see from the figure that the proposed attack requires less distortion to attack the selected examples, producing images that look visually better than the others. The advantage is particularly evident for the GTSRB case, but is still visible for the CIFAR-10 and MNIST images.

As for time complexity, we observe that though our attack aims at modifying fewer bits each time, its complexity is very similar to that of C&W attack. Specifically, if we allow 2000 iterations (10 binary searches) for each attack, for CIFAR-10, the new attack and C&W attack require about 800 seconds and 1000 seconds to attack an image, respectively (The times are measured using one NVIDIA RTX2080 GPU without paralleling.). On the other hand, LOTS is considerably faster since it needs about 80 seconds to attack an image. The reason behind the high computational complexity of C&W and our new attack is the binary search carried out at each step. In fact, we verified that, by reducing the number of steps the binary search consists of, the speed of both attacks improves greatly. However, since our main purpose is to test the robustness of the ECOC system, we did not pay much effort to optimize our attack from a computational point of view, all the more that its complexity is already similar to that of C&W attack.

As an overall conclusion, the experimental analysis reveals that, in the white-box scenario, the security gain that can be achieved through the ECOC scheme is quite limited since by properly applying existing attacks and especially by using the newly proposed attack, the ECOC classifiers could be attacked quite easily.

Another expected advantage of ECOC is that adversarial examples tend to be classified with a lower probability than

Input:
 The start point and number of binary search λ_1, n ;
 The step size and max iteration of gradient descent, ϵ, m ;
 To be attacked image, x ;

Output:
 Adversarial perturbation δ

```

(1) upper bound  $\leftarrow \infty$ 
(2) lower bound  $\leftarrow 0$ 
(3) for  $i \in [1, n]$  do
(4)  $\delta_1 \leftarrow 0$ 
(5) found adv  $\leftarrow$  False
(6)   for  $j \in [1, m]$  do
(7)     if  $x + \delta_j$  is adversarial and  $\|\delta_j\| < \|\delta\|$  then
(8)       found adv  $\leftarrow$  True
(9)     end if
(10)     $\delta_j \leftarrow \delta_j - \epsilon \times (\nabla_i / \|\nabla_i\|)$ , where  $\nabla_i$  is the gradient of equation (6) with current  $\lambda_i$  w.r.t. the perturbation  $\delta_j$ 
(11)  end for
(12) if found adv = True then
(13)   upper bound  $\leftarrow \lambda_i$ 
(14) else
(15)   lower bound  $\leftarrow \lambda_i$ 
(16) end if
(17) if upper bound =  $\infty$  then
(18)    $\lambda_i \leftarrow 10 \times \lambda_i$ 
(19) else
(20)    $\lambda_i \leftarrow (\text{upperbound} + \text{lowerbound})/2$ 
(21) end if
(22) end for

```

ALGORITHM 1: Solving optimization in (6).

TABLE 1: Results of the attack against ECOC for GTSRB classification.

Parameters	Proposed (ECOC)		C&W (ECOC)		C&W (one-hot)	
	ASR (%)	PSNR	ASR (%)	PSNR	ASR (%)	PSNR
($1e-4, 5, 100, 0$)	40.3	41.43	7.0	48.80	25.5	46.82
($1e-4, 5, 200, 0$)	45.6	41.58	8.6	48.48	37.5	45.73
($1e-4, 5, 500, 0$)	53.3	41.65	11.3	48.03	47.5	46.07
($1e-2, 5, 500, 0$)	70.6	39.85	20.0	43.94	61	43.41
($1e-1, 10, 2000, 0$)	93.3	38.97	42.3	39.20	81.5	42.51

Reported parameters indicate, respectively (start point, number of steps of binary search, max iterations, and confidence).

clean images. Here, we show that such a behavior can be inhibited, at the price of a slightly larger distortion, by increasing the confidence of the attack. If a larger confidence margin c is used, in fact, the model becomes more certain about the wrongly predicted class. To back such a claim, in Table 4, we report the results of the new attack for different confidence values c for the CIFAR-10 case (To clearly show the effect of confidence, we did not consider adversarial examples that do not reach the chosen confidence margin c , which leads to a slight drop of the ASR.). The table shows the average probability assigned by the ECOC model to the original class (Prob. true class) and to the target class of the attack (Prob. targ class), before and after the attack.

From the table, we see that, by increasing c , the adversarial examples are assigned higher and higher

probabilities for the target class, getting closer to those of the benign samples. In particular, the average probability for the target class passes from 0.546 (with $c = 0$) to 0.993 (with $c = 5$), which is even higher than the average probability of the clean images before the attack (0.908), and the probability of the original (true) class decreases from 0.194 (with $c = 0$) to a value lower than 0.001 (with $c = 5$). A similar behavior can be observed for the C&W attack when c is raised from 0 to 15.

Figure 4 shows the distribution of the probabilities assigned to the most probable class for clean and adversarial images generated by the proposed attack. The plot confirms that the ECOC classifier assigns low probabilities only in the presence of adversarial examples obtained with a low confidence value c . When c grows, in fact, the probability

TABLE 2: Results of the attack against ECOC for CIFAR-10 classification.

Parameters	Proposed (ECOC)		C&W (ECOC)		C&W (one-hot)	
	ASR (%)	PSNR	ASR (%)	PSNR	ASR (%)	PSNR
($1e-4$, 5, 100, 0)	69.3	38.59	53.6	39.71	92.5	40.03
($1e-4$, 5, 500, 0)	88.0	38.52	62.3	39.94	100	40.27
($1e-4$, 10, 200, 0)	90.6	37.84	79	37.32	100	40.18
($1e-4$, 10, 500, 0)	95.0	38.39	82.6	37.55	100	40.30
($1e-1$, 10, 2000, 0)	98.6	38.41	92.6	36.97	100	39.99

Reported parameters indicate, respectively (start point, number of steps of binary search, max iterations, confidence).

TABLE 3: Results of the attack against ECOC for MNIST classification.

Parameters	Proposed (ECOC)		C&W (ECOC)		C&W (one-hot)	
	ASR (%)	PSNR	ASR (%)	PSNR	ASR (%)	PSNR
($1e-3$, 10, 100, 0)	29.3	21.26	26	21.19	1.5	32.48
($1e-3$, 10, 200, 0)	43.6	21.49	35.6	20.73	8	27.69
($1e-3$, 10, 500, 0)	55.6	21.91	43.6	20.37	40.5	24.29
($1e-3$, 10, 1000, 0)	64.6	22.23	49	20.56	66.5	24.97
($1e-1$, 10, 2000, 0)	72.3	22.35	57.6	20.35	78	25.29

Reported parameters indicate, respectively (start point, number of steps of binary search, max iterations, confidence).

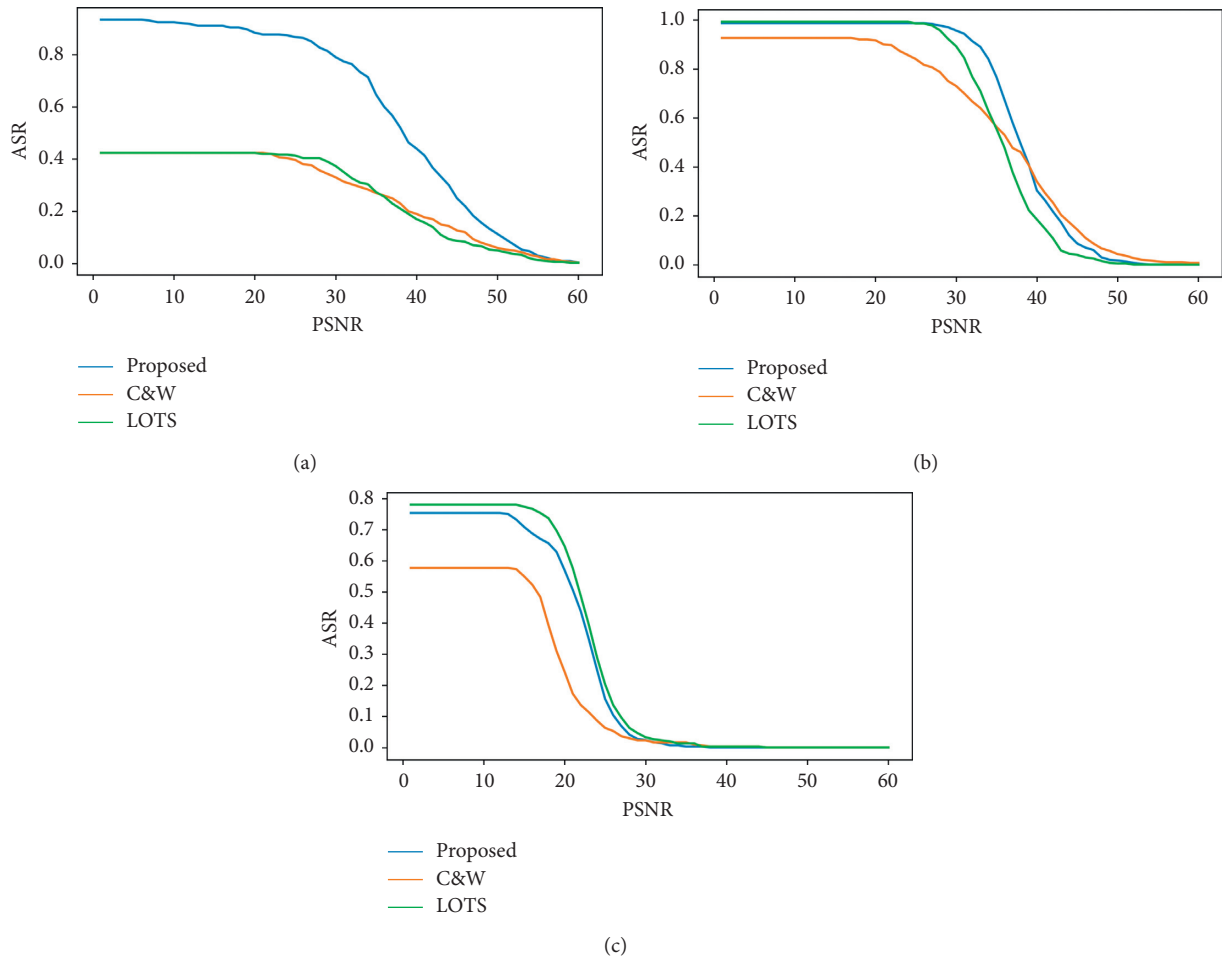


FIGURE 2: Performance of different attacks against the ECOC system. The x -axis indicates the PSNR(db) and y -axis indicates the ASR. (a) GTSRB. (b) CIFAR. (c) MNIST.

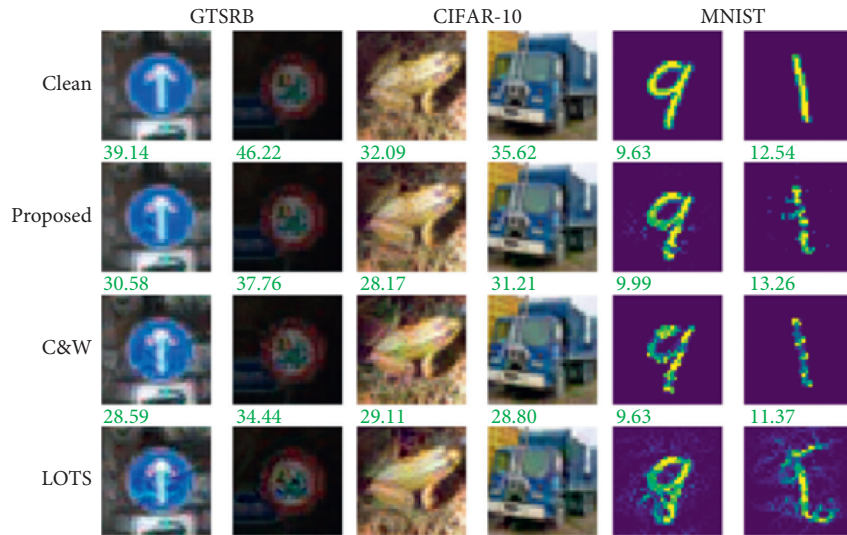


FIGURE 3: Examples of attacked images for different attacks. Besides the first row of clean images, the green number locates at the top left of each attacked image indicates its PSNR value.

TABLE 4: Output of probability values by the ECOC classifier on CIFAR-10 for different confidence margins of the attack.

C&W attack	$(1e-4, 5, 500, 0)$	$(1e-4, 5, 500, 8)$	$(1e-4, 5, 500, 12)$	$(1e-4, 5, 500, 14)$	$(1e-4, 5, 500, 15)$
ASR	62.3%	44.6%	44.6%	42.6%	42.3%
PSNR (dB)	39.94	40.78	39.57	39.00	38.40
Prob. true class	(B) 0.881 (A) 0.251	(B) 0.881 (A) 0.153	(B) 0.881 (A) 0.084	(B) 0.881 (A) 0.043	(B) 0.881 (A) 0.021
Prob. target class	(B) 0.013 (A) 0.328	(B) 0.013 (A) 0.534	(B) 0.013 (A) 0.721	(B) 0.013 (A) 0.843	(B) 0.013 (A) 0.914
Proposed attack	$(1e-4, 5, 500, 0)$	$(1e-4, 5, 500, 1.5)$	$(1e-4, 5, 500, 2.5)$	$(1e-4, 5, 500, 4.0)$	$(1e-4, 5, 500, 5.0)$
ASR	88.0%	87.6%	86.3%	85.1%	82.7%
PSNR (dB)	38.53	37.48	37.02	36.07	35.40
Prob. true class	(B) 0.908 (A) 0.194	(B) 0.908 (A) 0.063	(B) 0.908 (A) 0.024	(B) 0.908 (A) 0.005	(B) 0.908 (A) 0.001
Prob. target class	(B) 0.009 (A) 0.546	(B) 0.009 (A) 0.824	(B) 0.009 (A) 0.923	(B) 0.009 (A) 0.981	(B) 0.009 (A) 0.993

The parameters of the attacks are indicated according to the following format: (starting point, number of steps of binary search, max iterations, confidence). Prob. true and target class indicate the probabilities of the original (true) and target classes, before (B) and after (A) the attack.

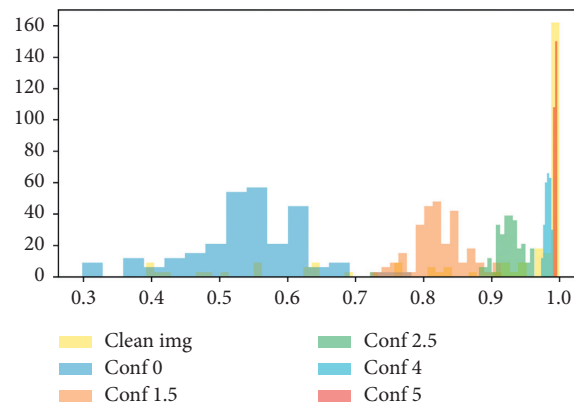


FIGURE 4: Distribution of probabilities assigned to the most probable class of the attacked examples with the proposed attack on CIFAR-10. The x -axis indicates the probability, and the y -axis indicates the number of examples classified with such a probability.

distribution of adversarial examples get closer and closer to that of clean images, and when $c = 5$, it becomes impossible to distinguish clean images and adversarial examples by setting a threshold on the probability assigned to the most probable class.

5. Related Works

Adversarial examples, i.e., small, often imperceptible, ad hoc perturbations of the input images, have been shown to be able to easily fool neural networks [1, 2, 5, 24] and have received great attention in the last years.

Different attacks have been proposed to obtain adversarial examples in various ways. Some works focus on diminishing the computational cost necessary to build the adversarial examples [2, 25], while others aim at lowering the perturbation introduced to generate the adversarial examples [1, 6, 26]. There are also some works whose goal is to find adversarial examples that modify only one pixel [27] and adversarial perturbations that can either fool several models at the same time [28], or can be applied to several clean images at the same time [4].

As a response to the threats posed by the existence of adversarial examples and by the ease with which they can be created, many defence mechanisms have also been proposed. According to [29, 30], defences can be roughly categorized into two branches, which either work in a reactive or proactive manner. The first class of defences is applied after the DNNs have been built. This class includes approaches exploiting randomization, like, for instance, stochastic activation pruning, in which node activations at each (or some) layers are randomly dropped out during the forward propagation pass [31], and, more recently, model switching [32], where random selection is performed between several trained submodels. Other approaches attempt to intentionally modify the network input to mitigate the adversarial perturbation, e.g., by projecting the input into a different space [33] or by applying some input transformations [34]. Other approaches attempt to reject input samples that exhibit an outlying behavior with respect to the unperturbed training data [35]. The second branch of defences aims at building more robust DNNs. One simple approach to improve the robustness against adversarial example is adversarial training, which consists in augmenting the training set with adversarial examples [10–12, 36]. More recently, as more attention has been paid to hidden layers with respect to the robustness of DNNs [37], it has been proven that rather than augmenting the training set, the robustness of DNNs can be improved by directly injecting adversarial noise into the hidden nodes, thus improving the robustness of single neurons [38, 39].

The ECOC scheme considered in this paper, belongs to the second class of defences and is derived from similar attempts made in the general machine learning literature to improve the robustness of multiclass classification problems [14, 40, 41]. The robustness of ECOC against adversarial examples is assessed in [15] by considering conventional adversarial attacks like [6, 42], which have not been explicitly designed for multilabel classification. As suggested in [17],

however, in order to properly assess the effectiveness of a defence mechanism, the case of an informed attacker should be considered, and then the robustness should be evaluated against attacks targeting the specific defence mechanism. Following the spirit of [17], in this paper, we developed a targeted attack against the ECOC system, which exploits the multiclass and multilabel nature of such a system. We observe that the capability of ECOC to hinder the generation of adversarial examples has already been challenged in [17] (Section 10); however, the analysis in [17] is carried out under the more favourable (for the attacker) assumption of a nontargeted attack, thus marking a significant difference with respect to the current work [4, 43].

6. Conclusion

In order to investigate the effectiveness of ECOC-based deep learning architectures to hinder the generation of adversarial examples, we have proposed a new targeted attack explicitly thought to work with such architectures. We measured the validity of the new attack experimentally on three common classification tasks, namely, GTSRB, CIFAR-10, and MNIST. The results we have got show the effectiveness of the new attack and, most importantly, demonstrate that the use of error correction to code the output of a CNN classifier does not increase significantly the robustness against adversarial examples, even in the more challenging case of a targeted attack. In fact, the ECOC scheme can be fooled by introducing a small perturbation into the images, both with the new attack and, to a lesser extent, by applying C&W and LOTS attacks with a proper setting. No significant advantage in terms of decision confidence is observed as well, given that, by properly setting the parameters of the attack, adversarial examples are assigned to the wrong class with a high probability.

Data Availability

The data used to support the findings of this study are available from the first author (bowenzhang.psnl@outlook.com) upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was partially supported by the China Scholarship Council (CSC) (file no. 201806960079).

References

- [1] C. Szegedy, W. Zaremba, I. Sutskever et al., “Intriguing properties of neural networks,” 2013, <https://arxiv.org/abs/1312.6199>.
- [2] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” 2014, <https://arxiv.org/abs/1412.6572>.

- [3] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: a survey," *IEEE Access*, vol. 6, pp. 14410–14430, 2018.
- [4] S. M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi et al., "Universal adversarial perturbations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1765–1773, Honolulu, HI, USA, July 2017.
- [5] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: a simple and accurate method to fool deep neural networks," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2574–2582, Las Vegas, NV, USA, June 2016.
- [6] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, San Jose, CA, USA, May 2017.
- [7] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597, San Jose, CA, USA, May 2016.
- [8] S. Gu and L. Rigazio, "Towards deep neural network architectures robust to adversarial examples," 2014, <https://arxiv.org/abs/1412.5068>.
- [9] U. Shaham, Y. Yamada, and S. Negahban, "Understanding adversarial training: increasing local stability of supervised models through robust optimization," *Neurocomputing*, vol. 307, pp. 195–204, 2018.
- [10] F. Tramèr, A. Kurakin, N. Papernot et al., "Ensemble adversarial training: attacks and defenses," 2017, <https://arxiv.org/abs/1705.07204>.
- [11] T. Miyato, A. M. Dai, and I. Goodfellow, "Adversarial training methods for semi-supervised text classification," 2016, <https://arxiv.org/abs/1605.07725>.
- [12] S. Zheng, Y. Song, T. Leung, and I. Goodfellow, "Improving the robustness of deep neural networks via stability training," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4480–4488, Las Vegas, NV, USA, June 2016.
- [13] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: bypassing ten detection methods," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 3–14, Dallas, TX, USA, November 2017.
- [14] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *Journal of Artificial Intelligence Research*, vol. 2, pp. 263–286, 1994.
- [15] G. Verma and A. Swami, "Error correcting output codes improve probability estimation and adversarial robustness of deep neural networks," *Advances in Neural Information Processing Systems*, pp. 8643–8653, 2019, <http://papers.nips.cc/paper/9070-error-correcting-output-codes-improve-probability-estimation-and-adversarial-robustness-of-deep-neural-networks>.
- [16] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.
- [17] F. Tramèr, "On adaptive attacks to adversarial example defenses," 2020, <https://arxiv.org/abs/2002.08347>.
- [18] A. Rozsa, M. Günther, and T. E. Boulton, "LOTS about attacking deep features," in *Proceedings of the 2017 IEEE International Joint Conference on Biometrics (IJCB)*, pp. 168–176, Denver, CO, USA, October 2017.
- [19] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: benchmarking machine learning algorithms for traffic sign recognition," *Neural Networks*, vol. 32, pp. 323–332, 2012.
- [20] Y. LeCun, C. Cortes, and C. J. C. Burges, "The MNIST database of handwritten digits," 1998, <https://yann.lecun.com/exdb/mnist/>.
- [21] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Multi-label vs. combined single-label sound event detection with deep neural networks," in *Proceedings of the 2015 23rd European Signal Processing Conference (EUSIPCO)*, pp. 2551–2555, Nice, France, 2015.
- [22] M. L. Zhang and Z. H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1338–1351, 2006.
- [23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, <https://arxiv.org/abs/1409.1556>.
- [24] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 427–436, Boston, MA, USA, June 2015.
- [25] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," 2016, <https://arxiv.org/abs/1607.02533>.
- [26] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroSecP)*, pp. 372–387, Saarbrücken, Germany, March 2016.
- [27] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.
- [28] C. Xie, Z. Zhang, Y. Zhou et al., "Improving transferability of adversarial examples with input diversity," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2730–2739, Long Beach, CA, USA, June 2019.
- [29] B. Biggio and F. Roli, "Wild patterns: ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317–331, 2018.
- [30] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: attacks and defenses for deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 9, pp. 2805–2824, 2019.
- [31] G. S. Dhillon, K. Azizzadenesheli, Z. C. Lipton et al., "Stochastic activation pruning for robust adversarial defense," 2017, <https://arxiv.org/abs/1803.01442>.
- [32] X. Wang, "Block switching: a stochastic approach for deep learning security," 2020, <https://arxiv.org/abs/2002.07920>.
- [33] A. Ilyas, A. Jalal, E. Asteri, C. Daskalakis, and A. G. Dimakis, "The robust manifold defense: adversarial training using generative models," 2017, <https://arxiv.org/abs/1712.09196>.
- [34] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: detecting adversarial examples in deep neural networks," 2017, <https://arxiv.org/abs/1704.01155>.
- [35] G. Tao, S. Ma, Y. Liu, and X. Zhang, "Attacks meet interpretability: attribute-steered detection of adversarial samples," *Advances in Neural Information Processing Systems*, pp. 7717–7728, 2018, <http://papers.nips.cc/paper/7998-attacks-meet-interpretability-attribute-steered-detection-of-adversarial-samples>.
- [36] H. Yu, A. Liu, X. Liu et al., "PDA: Progressive data augmentation for general robustness of deep neural networks," <https://arxiv.org/abs/1909.04839v3>.
- [37] M. Cisse, P. Bojanowski, E. Grave et al., "Parseval networks: improving robustness to adversarial examples," *Proceedings of*

- the 34th International Conference on Machine Learning*, vol. 70, pp. 854–863, 2017.
- [38] A. Liu, X. Liu, C. Zhang et al., “Training robust deep neural networks via adversarial noise propagation,” <https://arxiv.org/abs/1909.09034>.
 - [39] C. Zhang, A. Liu, X. Liu et al., “Interpreting and improving adversarial robustness of deep neural networks with neuron sensitivity,” <https://arxiv.org/abs/1909.06978>.
 - [40] N. Eghbali and G. A. Montazer, “Improving multiclass classification using neighborhood search in error correcting output codes,” *Pattern Recognition Letters*, vol. 100, pp. 74–82, 2017.
 - [41] M. Lachaize, S. L. Hégarat-Masclé, E. Aldea, A. Maitrot, and R. Reynaud, “Evidential framework for error correcting output code classification,” *Engineering Applications of Artificial Intelligence*, vol. 73, pp. 10–21, 2018.
 - [42] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” 2017, <https://arxiv.org/abs/1706.06083>.
 - [43] A. Athalye, N. Carlini, and D. Wagner, “Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples,” 2018, <https://arxiv.org/abs/1802.00420>.
 - [44] A. Athalye, N. Carlini, and D. Wagner, “Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples,” 2018, <https://arxiv.org/abs/1802.00420>.
 - [45] F. Tramèr, “On adaptive attacks to adversarial example defenses,” 2020, <https://arxiv.org/abs/2002.08347>.