Exploiting the Sensitivity of L_2 Adversarial Examples to *Erase-and-Restore*

Fei Zuo
University of South Carolina
Columbia, SC, USA
fzuo@email.sc.edu

Qiang Zeng
University of South Carolina
Columbia, SC, USA
zeng1@cse.sc.edu

ABSTRACT

By adding carefully crafted perturbations to input images, adversarial examples (AEs) can be generated to mislead neural-network-based image classifiers. L_2 adversarial perturbations by Carlini and Wagner (CW) are among the most effective but difficult-to-detect attacks. While many countermeasures against AEs have been proposed, detection of adaptive CW-L_2 AEs is still an open question. We find that, by randomly erasing some pixels in an L_2 AE and then restoring it with an inpainting technique, the AE, before and after the steps, tends to have different classification results, while a benign sample does *not* show this symptom. We thus propose a novel AE detection technique, Erase-and-Restore (E&R), that exploits the intriguing sensitivity of L_2 attacks. Experiments conducted on two popular image datasets, CIFAR-10 and ImageNet, show that the proposed technique is able to detect over 98% of L_2 AEs and has a very low false positive rate on benign images. The detection technique exhibits high transferability: a detection system trained using CW-L_2 AEs can accurately detect AEs generated using another L_2 attack method. More importantly, our approach demonstrates strong resilience to adaptive L_2 attacks, filling a critical gap in AE detection. Finally, we interpret the detection technique through both visualization and quantification.

CCS CONCEPTS

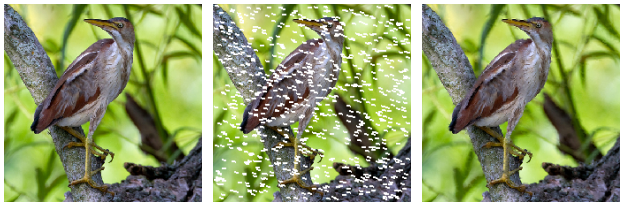
• **Security and privacy → Software and application security**;
• **Computing methodologies → Machine learning**.

KEYWORDS

adversarial example; adversarial detection; image classification

ACM Reference Format:
Fei Zuo and Qiang Zeng. 2021. Exploiting the Sensitivity of L_2 Adversarial Examples to *Erase-and-Restore*. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security (ASIA CCS '21), June 7–11, 2021, Hong Kong, Hong Kong*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3433210.3437529

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ASIA CCS '21, June 7–11, 2021, Hong Kong, Hong Kong
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8287-8/21/06…$15.00
https://doi.org/10.1145/3433210.3437529



(a) Original image (b) Corrupted image (c) Restored image

Figure 1: Restoring lost parts of an image with inpainting.

1 INTRODUCTION

By adding deliberately crafted perturbations into an image, an attacker is able to create an *adversarial example* (AE), which misleads a neural-network-based classifier to output an incorrect prediction result. Worse, the malicious perturbations in an AE are so subtle that they are usually human-imperceptible. As neural networks are increasingly deployed, AEs raise crucial security concerns especially in many vision-related applications.

The term *adversarial example* can be formally defined as follows. For a pre-trained DNN f, let x be an original image. An adversarial example x^{adv}, derived from x, can guide the model f to make an incorrect prediction. Moreover, to hide the adversarial perturbation, the generation of x^{adv} is equivalent to solve the following constrained optimization problem:

$$\begin{aligned} &\min_{x^{adv}} \ \|x^{adv} - x\|_p \\ &\text{s.t. } \ y' = f(x^{adv}),\ y = f(x),\ \text{and } y \neq y' \end{aligned} \tag{1}$$

where y and y' are respectively the prediction results of feeding x and x^{adv} to f.

To gauge such adversarial perturbations, L_p norms are usually used to quantitatively describe the discrepancy between x and x^{adv}. According to the value of p in Equation 1, the mainstream AE generation algorithms can be categorized into three families: L_0, L_2 and L_∞ attacks. Informally, L_0 measures the number of modified pixels, L_2 the Euclidean distance between x and x^{adv}, and L_∞ the largest modification among all the modified pixels.

As suggested by Carlini and Wagner [7], defenders should consider evaluating "*a powerful attack*" and particularly emphasized L_2 attacks (Section 9 in [7]). Other researchers also agree that L_2 attacks by Carlini and Wagner (CW) [7] "*are among the most effective white-box attacks and should be used among the primary attacks to evaluate potential defences*"[41]. Although researchers have proposed many AE detection methods [31, 37, 38, 52], recent studies [5, 6, 26] show that the detection usually goes ineffective

when facing adaptive CW-L_2 AEs. Thus, how to accurately detect adaptive L_2 AEs is still an open question. We focus on tackling L_2 AEs in this work, and our goal is a technique that not only detects L_2 AEs accurately but is also resilient to adaptive attacks.

We have two key insights. First, we observe that those deliberately corrupted pixels exert a malicious influence *altogether* (e.g., through multiple rounds of optimizations during AE generation). It implies that a destruction of the completeness of the influence by the perturbed pixels can cause a failure of the attack. Second, while destruction may also harm the classification accuracy for benign samples, there exist very effective *inpainting* techniques [36, 48, 49] in the image processing area that can help restore a partially corrupted image. For example, Figure 1(a) shows an original image, and Figure 1(b) a corresponding corrupted image where many regions are erased. After inpainting, as shown in Figure 1(c), the corrupted image is well restored.

Thus, we hypothesize that if we ***randomly*** erase a portion of pixels from an AE and then apply inpainting to it, the attack will probably fail for two reasons. Discarding many small regions from an AE will ruin the holistic adversarial influence formed by the maliciously perturbed pixels. Second, the inpainting typically restores the image in a benign way that does *not* preserve the malicious influence. By contrast, if we apply the same "*Erase-and-Restore*" (E&R) operations to a benign sample, the classification results, before and after the steps, tend to be similar, as inpainting by design is to reverse deterioration of benign images.

Figure 2 illustrates our insights and observations using six color images from CIFAR-10. A *random mask* (mask, for short) in our work describes the locations of pixels that are randomly erased. We randomly erase 5% of the pixels of each image. The AEs are generated using the CW algorithm [7]. As shown in Figure 2(a), the classification results of each AE, before and after the E&R operations, are different. By contrast, as shown in Figure 2(b), the classification results of each benign sample, before and after the steps, are the same. Our large-scale experiments (Section 4) also show consistent results.

We consider the sensitivity to E&R operations as an exploitable characteristic of L_2 AEs, and propose a novel AE detection technique: given an image, if the classification results before and after E&R vary greatly, it is an AE; otherwise, a benign sample. We accordingly implement an L_2 AE detector, named Themis. To improve the detection accuracy, it is enhanced by applying E&R multiple times. Specifically, given an image I_0, we *randomly* erase some pixels of I_0 each time to create a sequence of images $\{I_1, I_2, \cdots, I_n\}$. Next, an inpainting technique is applied to them to obtain the restored images $\{I'_1, I'_2, \cdots, I'_n\}$. Finally, a classifier makes use of the prediction results of I_0 and the restored images to determine whether I_0 is an AE.

We have evaluated our system using the popular image datasets CIFAR-10 and ImageNet. Two widely-discussed L_2 AE generation methods, CW [7] and DeepFool [40], are considered in the evaluation. Our experiments show that the proposed detection technique is very effective. Take the CW [7] attack as an example, on the CIFAR-10 dataset, Themis can detect 100% AEs with a false positive rate (FPR)=0, and on ImageNet, it can detect 99.3% AEs with FPR = 2.7%. In addition, the detection technique demonstrates three notable characteristics. ❶ It is **target-model agnostic**: a detector trained using AEs targeting one neural network model can be directly used to detect AEs targeting another. ❷ It has good **transferability**: a detector trained using AEs generated by one attack method can be directly used to detect AEs by another. ❸ More importantly, it shows **high resilience to adaptive attacks**. Finally, we interpret the effectiveness of the detection technique through both visualization and quantification.

The key contributions of our work include:

- We find an interesting characteristic of L_2 AEs, whose classification results vary sharply when Erase-and-Restore operations are applied; meanwhile, benign samples are not so sensitive.
- We propose to exploit the characteristic for AE detection, and employ the idea of sampling to enhance the detection. By applying E&R for multiple times, richer features are generated to improve the detection accuracy.
- We implement the detection technique in Themis and evaluate it on two popular datasets, CIFAR-10 and ImageNet. The experiment results show that Themis outperforms prior techniques (such as NIC [33], LID [34], and Feature Squeezing [52]), achieving not only **the highest detection rate** but also the **lowest false positive rate**. We are to make the source code, datasets, and models of this work publicly available.[1]
- The detection technique is target-model agnostic and shows high transferability across different L_2 attack methods. Furthermore, it demonstrates strong resilience to adaptive CW-L_2 attacks, filling a critical gap in AE detection.
- We interpret the effectiveness of the detection technique in multiple ways.

2 BACKGROUND AND THREAT MODEL

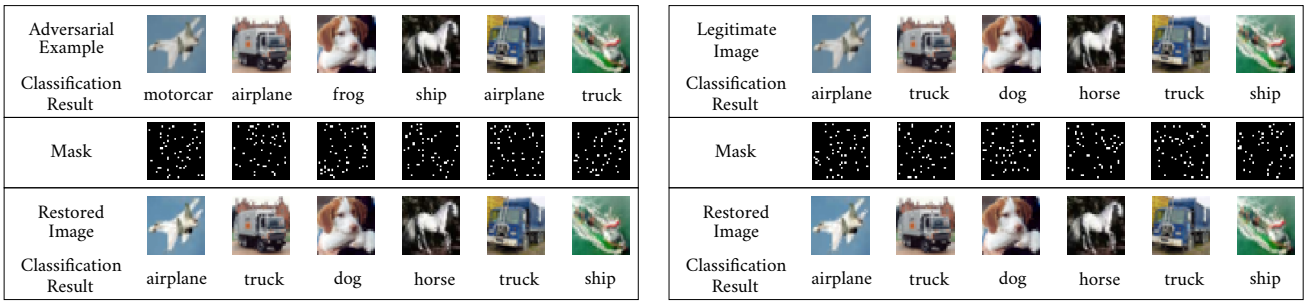
2.1 Attack Algorithms

Adversarial attacks can be categorized as either non-targeted or targeted ones. The aim of a non-targeted attack is to make the input be classified as any arbitrary class except the correct one. By contrast, the aim of a targeted attack is a specific attacker-desired incorrect result, which is more threatening. Next, we briefly describe the two most popular L_2 AE generation methods.

Carlini & Wagner Attacks Carlini and Wagner [7] designed a group of targeted AE generation methods which are denoted as CW attacks. According to the distance metrics adopted in an optimization target, CW attacks can be divided into three types: L_0-, L_2- and L_∞-norm. In this paper, we mainly examine *CW-L_2 attacks*, which are the most difficult to detect [5, 26].

Due to a few creative designs, the CW attacks achieve performance superior to other attack methods. The first and foremost innovative design is using a logits-based objective function rather than softmax-cross-entropy loss, which plays a key role in the resilience improvement of the attack against defensive distillation [42]. Secondly, this algorithm maps the target variable to a space of the inverse trigonometric function, so that the problem is suitable to be solved by a modern optimizer, e.g. Adam [28]. Finally,

[1] https://github.com/quz105/Erase-and-Restore.



(a) Adversarial examples **(b) Benign samples**

Figure 2: Different impacts of "Erase-and-Restore" on AEs and benign samples.

a *confidence-level* parameter κ is introduced; as κ increases, the model classifies the resulting AE as the attacker-desired label more likely, giving the attacker flexibility to make a trade-off between the degree of perturbations and misclassification probability.

DeepFool Moosavi et al. [40] developed the DeepFool attack that is used to create non-targeted AEs. The algorithm utilizes an iterative linearization of the classifier to generate L_2 minimization-based perturbations. To simplify the problem, the neural networks are imagined to be linear, so that the decision boundaries are a set of hyper-planes. Consequently, a polyhedron can be used to describe the output space. Assuming that f is a binary differentiable classifier, to mislead the decision of f near the current point x_i, the minimal perturbation is the orthogonal projection of x_i onto the separating hyper-plane. At each iteration the minimal perturbation of the linearized classifier is computed as

$$\arg\min_{\delta_i} \|\delta_i\|_2 \quad \text{s.t. } f(x_i) + \nabla f(x_i)^T \delta_i = 0 \tag{2}$$

where δ_i is the perturbation imposed on x_i. Note that neural networks are not actually linear, so the search is repeated until a successful AE is found.

2.2 Threat Model

The adversary has full knowledge of the target model (including both its architecture and parameters). He also knows the existence and internal details of the detector, and is allowed to *adapt attacks*. In adaptive attacks, the attacker tries to fool the image classifier and the detector at the same time. We consider adaptive attacks and evaluate the resilience of our detector to them in Section 6.

3 EXPERIMENTAL SETUP

Before presenting our defense scheme, we introduce the image datasets and the corresponding target neural networks on which we verify our key insights and evaluate the proposed approach.

Image datasets. We generate AEs using two popular datasets: CIFAR-10 and ImageNet, both of which are widely used in image classification tasks. In particular, for ImageNet, we adopt the *ILSVRC2012* samples to keep consistent with the prior state-of-the-art AE detector [33].

Target neural network models. (1) For CIFAR-10, we use two neural networks as the target models: a 32-layered ResNet model [25] (denoted as *ResNet32*), and a model structure described in [7] (denoted as *Carlini*). We train these two target neural network models from scratch (the accuracies of the two models are 91.96% and 78.86%, comparable with those published in prior works [33, 52]). (2) For ImageNet we re-use a 50-layered ResNet model [25] provided in Keras [9] (denoted as *ResNet50*).

AE generation and data preparation. Like existing AE detection works, only images that are correctly classified by the corresponding target model are used to generate AEs in our experiments. To generate *targeted* AEs, we designate the *next* class as the target class, similar to many other AE detection works [33, 52, 55]. Only AEs that can successfully fool the target models are used in the evaluation. For ImageNet, we collect 30,000 legitimate images and create 30,000 AEs: DeepFool and CW-L_2 generate 15,000 AEs each. The number of CW-L_2 AEs with each given confidence level (i.e.,κ= 0.0, 0.4, and 1.0) is the same, that is 5,000 for each sub-group. In the dataset, 80% of instances are used for training and the remaining 20% for testing, denoted as $\mathcal{D}_I$-`Train` and $\mathcal{D}_I$-`Test`, respectively. Similarly, for CIFAR-10, based on the types of target model, we have four dis-joint datasets, $\mathcal{D}_C$-`Carlini-Train`, $\mathcal{D}_C$-`Carlini-Test`, $\mathcal{D}_C$-`ResNet-Train`, and $\mathcal{D}_C$-`ResNet-Test`. The former two and the latter two datasets have the same size and data composition as $\mathcal{D}_I$-`Train` and $\mathcal{D}_I$-`Test`, respectively. All AEs are generated using the opensource tool `Foolbox` [45].

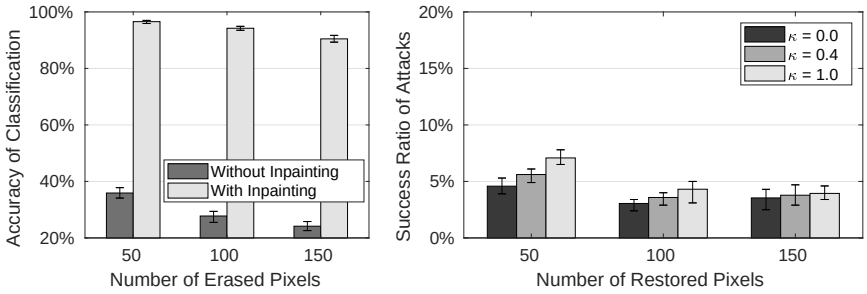
Inpainting algorithm. The inpainting algorithm we choose in this work is designed by Telea [49]. This inpainting algorithm needs to solve an *Eikonal* equation, which is rarely differentiable everywhere. Considering the inpainting algorithm is *not* fully differentiable, it results in a non-negligible obstacle for adaptive attackers.

The experiments were performed on a computer running the Ubuntu 18.04 operating system with a 64-bit 3.6 GHz Intel® Core(TM) i7 CPU, 16 GB RAM and a GeForce® GTX 1070 GPU.

4 THE PROPOSED APPROACH

4.1 Our Insights

Effects of erasing (or adding noises) alone. Due to the optimization nature of AE generation methods like CW and DeepFool,



(a) Benign samples **(b) Success rates of AEs with E&R**

Figure 3: Impacts of E&R on benign samples and AEs.

maliciously manipulated pixels in an AE are deliberately selected and perturbed. Thus, each of the perturbed pixels plays a certain role in the attack. By *randomly erasing* many pixels of an input image, it is likely to corrupt some of the perturbed pixels or their surrounding pixels in an AE, rendering the attack ineffective.

In the case of *benign* samples, however, the erasing operation, which is equivalent to introducing random noises to images, will significantly degrade the accuracy of the classifier. The close correlation between the image quality and the accuracy of image classification has been widely studied in previous works [12, 15, 16]. They mention that neural networks are susceptible to random noise distortions. For example, Costa et al. [12] point out that *"noises can hinder classification performance considerably and make classes harder to separate."*

Combining erasing and inpainting. We thus propose to apply *inpainting* after the erasing operation. Inpainting is a category of techniques for restoring damaged regions of images. Given an erased region, an inpainting technique infers and recovers its original pixels. *Our insight* is that, while inpainting works very well for recovering benign samples, its recovering effect is usually *not* what the AE attacker desires, as the maliciously perturbed regions, once erased, can hardly be recovered to the attacker-intended values.

We further design experiments to verify the two insights in Section 4.2.

4.2 Verifying Our Insights

From CIFAR-10, we randomly select 1,000 images that can be correctly classified by *ResNet32*. As shown in Figure 3(a), after randomly erasing 50~150 (around 5%~15%) of the pixels in each image, without inpainting, the classification accuracy significantly degrades from 100% to the range from 24.2% (when erasing 15%) to 35.9% (when erasing 5%), which verifies that erasing alone harms the classification accuracy for benign images significantly. By contrast, with inpainting applied, the classification accuracy recovers to 90.5%~96.6%.

Besides, for each benign image we use the CW algorithm to generate three AEs with three different confidence levels ($\kappa = 0.0$, 0.4, and 1.0, respectively). All the AEs successfully fool the *ResNet32* model. As shown in Figure 3(b), after randomly erasing 50~150 (around 5%~15%) of the pixels in each AE and then restoring them using inpainting, the success rate of attacks dramatically decreases from the original 100% to the range 3.1%~7.1%.

Similar results can be observed on the ImageNet dataset as well. (1) Specifically, we randomly select 1,000 images from ImageNet that can be correctly classified by the *ResNet50* model. For example, after erasing and restoring 5% of the pixels in each image, the classification accuracy stays at 96.3%. (2) On the other hand, when we apply the same erasing and restoring operations to the 1,000 AEs generated from these benign images, the success rate of attacks decreases from 100% to around 4.1%.

Therefore, it can be concluded that E&R has very small impacts on benign samples, but large impacts on AEs, demonstrating a noticeable contrast.

4.3 Approach Details

Based on our insights, we propose a novel AE detection technique, named E&R, that exploits the sensitivity of AEs to E&R operations, and implement it in a system, called Themis, as shown in Figure 4. (1) Given an input image I_0, we ***randomly erase*** λ ***pixels*** of it to create a deteriorated image I. Employing the idea of sampling, this step is repeated for n times to obtain a sequence of deteriorated images $\{I_1, I_2, \cdots, I_n\}$. The *intuition* behind it is that even if an AE "luckily" evades the detection once, it is very unlikely for it to hide itself throughout the multiple samples. (2) Next, an inpainting technique is leveraged to produce a corresponding sequence of ***restored*** images $\{I'_1, I'_2, \cdots, I'_n\}$. (3) Finally, we feed both the input image I_0 and $\{I'_1, I'_2, \cdots, I'_n\}$ into a neural-network classifier, and collect all the classification results.

Given an image in CIFAR-10, its classification result is a vector $\in \mathbb{R}^{10}$ (since there are 10 classes in the dataset). We simply concatenate all the classification-result vectors for both I_0 and $\{I'_1, I'_2, \cdots, I'_n\}$ to obtain a feature vector $\in \mathbb{R}^{10\times(n+1)}$ for training the AE classifier.

Given an image from the ImageNet, its classification result is a vector $\in \mathbb{R}^{1000}$ (since there are 1,000 classes in the dataset). Thus, the number of features to be fed to our classifier is $1000 \times (n+1)$, which is too large. To make the training of our classifier more feasible, Principal Component Analysis (PCA) is performed on the classification results of I_0 and $\{I'_1, I'_2, \cdots, I'_n\}$, to reduce the dimensionality to a lower value d. Unless otherwise specified, we set d to 10 (1% of the original dimensionality) to keep consistent with CIFAR-10. Note that the number of principal components

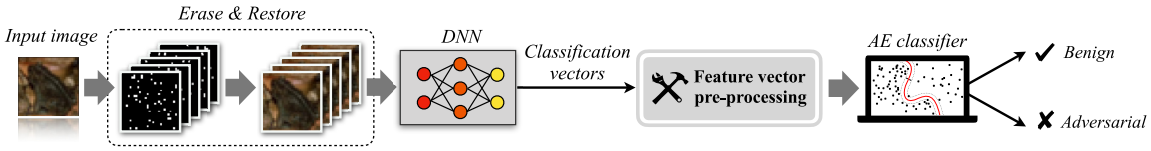


Figure 4: Architecture of Themis.

Table 1: Performance of Themis. After Themis is trained using training datasets that contain benign samples, CW and DeepFool AEs, the detection rate and FPR (the rate of benign samples misclassified as AEs) are measured using testing sets.

Dataset	Target Model	Classifier	FPR	Detection Rate: CW-L_2			Detection Rate: DeepFool
				κ=0.0	κ=0.4	κ=1.0	
CIFAR-10	Carlini	SVM	0.6%	100%	100%	100%	99.4%
		AdaBoost	0.0%	100%	100%	100%	98.3%
	ResNet32	SVM	2.8%	99.4%	99.6%	99.6%	99.8%
		AdaBoost	0.9%	99.4%	99.2%	99.4%	99.8%
ImageNet	ResNet50	SVM	3.5%	97.9%	98.4%	98.7%	93.7%
		AdaBoost	2.7%	98.9%	99.2%	99.3%	95.0%

should be less than both the number of features and the number of samples, when solving PCA based on the truncated SVD (singular value decomposition). In our case, the number of samples is $n+1$; we thus let $n = 11$ (we discuss the impact of n's values with detailed experimental results in Section 5.3). We concatenate the vectors of principal components for both I_0 and $\{I'_1, I'_2, \cdots, I'_n\}$ to obtain a feature vector for training our classifier.

The value of the parameter λ (number of pixels to be erased) is set to 10% of the pixels in an input image. We adopt this value for two reasons. (1) As shown in Figure 3, when 10% of the pixels are erased and restored, it harms the success rate of AEs most heavily, without degrading the classification accuracy for benign samples significantly. (2) The inpainting algorithm we adopt performs very well when the portion of corrupted pixels in an image is less than 15% [49].

It is worth mentioning that $\lambda = 10\%$ leads to an enormous randomness pool. Take an image in CIFAR-10 as an example, the size of which is 32×32: with λ=100 ($\approx$ 10% of the pixels), the number of unique masks is around 7.7×10^{140}. It is thus very unlikely for an adaptive attacker to correctly predict which masks will be used by our detector.

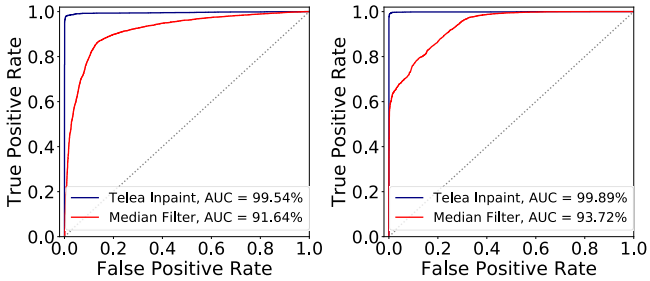
We train our AE classifier using two supervised learning techniques: *AdaBoost* [20] and *SVM* [11].

5 EVALUATION

We evaluate the detection performance of the proposed scheme against L_2 attacks in terms of *detection rate* and *false positive rate* (FPR). The detection rate is defined as the ratio of the number of successfully detected AEs to the total number of AEs. FPR refers to the fraction of benign samples that are misclassified as AEs.

5.1 Detection Performance

We use $\mathcal{D}_I$-`Train`, $\mathcal{D}_C$-`Carlini-Train`, and $\mathcal{D}_C$-`ResNet-Train` (see Section 3) to train our detectors and evaluate them based on the corresponding testing sets.



(a) SVM **(b) AdaBoost**

Figure 5: ROC curves.

CW-L_2 attacks. As shown in Table 1, the proposed technique achieves very high detection rates (up to 100% on CIFAR-10, and 99.3% on ImageNet) with low FPR values. The results are stable across different target models, confidence levels, and classification methods.

In addition to SVM and Adaboost, we also train a fully connected neural network as the AE classifier, and obtain very similar results. It shows that it does not affect the performance by using a more sophisticated classifier. It also indicates that the effect of E&R does not depend on a specific classifier type.

DeepFool attacks. For another leading L_2 AE generation algorithm—DeepFool (see Section 2.1), we observe very similar results as CW-L_2. Table 1 shows that our detector achieves very high detection rates (up to 99.8% on CIFAR-10, and 95.0% on ImageNet) with low FPR values.

Comparison with baseline. To illustrate the the benefits of the Telea inpainting algorithm used in our detector, we compare it with a baseline method, which uses a median filter to recover the damaged pixels. In particular, the window size of our median filter is

Table 2: Comparison with other AE detectors (DR: Detection Rate). We use the same attack settings as used in prior work [33, 52].

Dataset	CIFAR-10				ImageNet			
Detector	THEMIS	NIC	FS	LID	THEMIS	NIC	FS	LID
FPR	0.6%	4.2%	5.6%	4.9%	2.7%	14.6%	8.3%	14.5%
DR: CW-L_2	100%	96%	100%	86%	98.9%	96%	92%	78%
DR: DFool	99.4%	91%	77%	84%	95.0%	92%	79%	83%

3×3, which is also adopted by Feature Squeezing [52]. Without loss of of generality, the datasets we use are $\mathcal{D}_C$-ResNet-Train and $\mathcal{D}_C$-ResNet-Test. We replace the Telea inpainting with the median filter in our implementation to build a baseline detector. Figure 5 shows the comparison result using ROC (receiver operating characteristic) curves of the different detectors. As shown in Figure 5(a), when SVM is used as the classifier, the AUC value declines from 99.54% to 91.64%. Similarly, as shown in Figure 5(b), when AdaBoost is used, the AUC value correspondingly declines from 99.89% to 93.72%. Thus, a high-quality inpainting method is closely related to the final performance of our AE detector.

Comparison with prior work. As summarized in Table 2, we compare THEMIS with some state-of-the-art AE detectors—NIC [33], LID [34], and Feature Squeezing [52]. For CW-L_2 attack, their experiments only examine κ = 0.0, which is the default setting, so we also list the results under κ = 0.0 in Table 2 (see Table 1 for the results of our detector under other κ values). We take NIC as an example here. With respect to CIFAR-10, NIC obtains the detection rate 96% (see Table I in [33]), while our system achieves the detection rate **100%**. With respect to ImageNet, the detection rate of NIC is 96% (see Table I in [33]), while our detection rate is **98.9%**. In terms of DeepFool, THEMIS also outperforms other AE detectors. When considering CIFAR-10, our system obtains the detection rate **99.4%**, while NIC [33] obtains the detection rate 91.0% (see Table I in [33]). Similarly, when considering ImageNet, THEMIS can achieve the detection rate **95.0%**, that is superior to NIC, the detection rate of which is 92%.

More importantly, from the angle of FPR, the performance of THEMIS is significantly better than other detectors. For example, when considering CIFAR-10, the FPR of NIC is 4.2%, while ours is **0.6%**. Moreover, when considering ImageNet, the FPR of NIC is 14.6%, while ours is only **2.7%**. It is worth noting that the distribution of adversarial and benign images is not balanced in practice—most inputs should be benign. Thus, FPR is a very important metric to evaluate the model performance: a lower FPR indicates that the system makes fewer mistakes for benign images. THEMIS is able to keep both a high detection rate and a *very low FPR*.

5.2 Notable Characteristics

Target-model agnostic. We are interested in finding out whether a detector trained using AEs targeting one model can be directly used to detect AEs targeting another—that is, whether it is *target-model agnostic*. We thus train our system using CW-L_2 AEs in $\mathcal{D}_C$-Carlini-Train, and test it using CW-L_2 AEs in $\mathcal{D}_C$-ResNet-Test.

Table 3: Target-model agnostic property of THEMIS.

Target Model (Train → Test)	Classifier	Detection Rate		
		κ=0.0	κ=0.4	κ=1.0
Carlini→ResNet32	SVM	100%	100%	100%
	AdaBoost	97.9%	97.9%	98.2%
ResNet32→Carlini	SVM	99.9%	99.9%	99.8%
	AdaBoost	99.7%	99.8%	99.6%

As Table 3 shows, the detection rate is as high as 100%. We then train the system using CW-L_2 AEs in $\mathcal{D}_C$-ResNet-Train, and test it using CW-L_2 AEs in $\mathcal{D}_C$-Carlini-Test; the detection rate is as high as 99.9%.

Therefore, this experiment not only confirms that THEMIS is *target-model agnostic*, but also demonstrates that THEMIS has low risk of overfitting.

Transferability. We are also interested in the transferability of our detector—whether THEMIS trained on one type of AEs can be directly applied to detect another type of AEs that are *unseen* during training. To verify it, we *train* THEMIS using CW-L_2 AEs in $\mathcal{D}_C$-Carlini-Train, without loss of generality. Then, we test the trained system using DeepFool AEs in $\mathcal{D}_C$-ResNet-Test and $\mathcal{D}_C$-Carlini-Test , and our system can achieve detection rates 97.1% and 96.2%, respectively. Thus, we can conclude the proposed technique has very good transferability, that is, it keeps effective in handling unseen AE generation methods.

Explanation. The two notable properties of THEMIS—target-model agnostic and good transferability—can be attributed to the unique advantage of the proposed approach: benign samples and AEs show distinct sensitivities to the E&R operations, which do not depend on the target model and the attack method.

5.3 Value Selection for the Parameter n.

We use n = 11 in the previous experiments. Here, we investigate the impacts of different values of n on the detector's performance. The CW-L_2 AEs in $\mathcal{D}_C$-Carlini-Train, $\mathcal{D}_C$-ResNet-Train, and $\mathcal{D}_I$-Train are used in this experiment. For CIFAR-10, which has only 10 classes (thus no PCA is needed), varying the value of n has little impacts. However, for ImageNet, the value of n has noticeable impacts: when n increases, the AE detection rate increases and FPR decreases (see Table 4 and Table 5 for more details). The reason is that by increasing n, more principal components can be extracted (see Section 4). However, when $n > 11$, the performance improvement is negligible, probably because the extra principal components

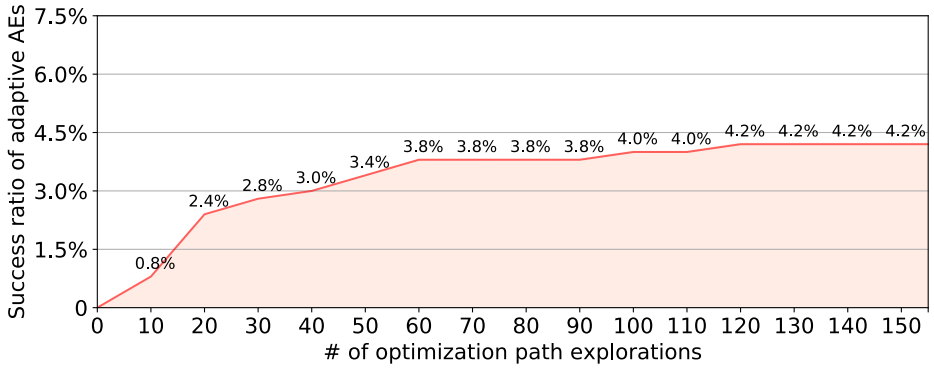


Figure 6: Success ratio of adaptive AEs.

Table 4: Impacts of different values of n (CIFAR-10).

Target Model	Classifier	FPR	Detection Rate			
			κ=0.0	κ=0.4	κ=1.0	
Carlini	SVM	0.4%	100%	100%	100%	n=3
	Adaboost	0.0%	100%	100%	99.9%	
ResNet32	SVM	3.6%	99.6%	99.6%	99.6%	
	Adaboost	0.9%	99.2%	99.1%	98.5%	
Carlini	SVM	0.4%	100%	100%	100%	n=5
	Adaboost	0.0%	100%	99.9%	99.9%	
ResNet32	SVM	3.3%	99.6%	99.6%	99.6%	
	Adaboost	0.7%	99.2%	99.1%	99.0%	
Carlini	SVM	0.4%	100%	100%	100%	n=7
	Adaboost	0.0%	100%	100%	99.8%	
ResNet32	SVM	2.9%	99.6%	99.7%	99.7%	
	Adaboost	0.9%	99.3%	99.1%	98.9%	
Carlini	SVM	0.4%	100%	100%	100%	n=9
	Adaboost	0.0%	100%	100%	99.8%	
ResNet32	SVM	3.0%	99.7%	99.7%	99.7%	
	Adaboost	0.7%	99.3%	99.3%	99.1%	
Carlini	SVM	0.4%	100%	100%	100%	n=11
	Adaboost	0.0%	99.8%	99.9%	99.7%	
ResNet32	SVM	2.8%	99.6%	99.7%	99.7%	
	Adaboost	0.8%	99.0%	99.2%	98.9%	

Table 5: Impacts of different values of n (ImageNet).

Target Model	Classifier	FPR	Detection Rate			
			κ=0.0	κ=0.4	κ=1.0	
ResNet50	SVM	9.8%	95.4%	95.1%	95.5%	n=3
	Adaboost	6.6%	93.1%	91.4%	93.8%	
	SVM	4.7%	95.5%	95.8%	97.3%	n=5
	Adaboost	2.8%	96.5%	97.6%	97.2%	
	SVM	3.6%	97.6%	98.1%	98.2%	n=7
	Adaboost	2.1%	97.9%	98.6%	98.6%	
	SVM	3.5%	97.6%	98.0%	98.3%	n=9
	Adaboost	2.0%	98.0%	98.4%	98.8%	
	SVM	3.2%	97.6%	98.1%	98.5%	n=11
	Adaboost	1.4%	98.4%	98.5%	98.9%	

do not provide useful features for AE detection. Therefore, we adopt $n = 11$.

5.4 Efficiency of Themis

We investigate the efficiency of the proposed technique on ImageNet because large-sized images consume more processing time. For a single image, ResNet50 needs approximately 1.076 seconds for classification. Since parallel computing is supported by GPU, given a relatively small number of images as inputs (e.g., $n = 11$), it takes similar time to generate the classification vectors for them. Apart from this, to detect AE, our method brings additional 1.01 seconds by average. In detail, it consumes 0.264 seconds for the inpainting, 0.744 seconds for the PCA-based dimension reduction, and 0.002 seconds for the final prediction (taking SVM as an example). In short, our detector causes a small delay.

6 RESILIENCE TO ADAPTIVE ATTACKS

In an adaptive attack threat model, an adversary knows the existence and internal details of our detector and *adapts* the attacks to bypass the detection. We thus seek to study the resilience of Themis to adaptive attacks.

An AE detector can be categorized as either differentiable or non-differentiable. Several previous works propose defense mechanisms that apply differentiable transformations to an image before detection or classification [21, 22, 38, 50]. But attackers can circumvent these differentiable defenses by "*differentiating through them*"—*i.e.,* by taking the gradient of a class probability regarding input pixels through both the CNN and the transformation [5, 26, 43]. This strategy, however, is *inapplicable* to bypassing Themis. Due to the random-erasing and inpainting-based restoring, our approach is not only non-differentiable but involves tremendous randomness.

To bypass non-differentiable defences, Backward Pass Differentiable Approximation (BPDA) is proposed [2]. To handle defenses that employ randomized transformation to the input (like ours), it applies Expectation over Transformation [3] to compute the gradient over the expected transformation to the input. However, in our approach the erased pixels are randomly selected among all the image pixels, and there are around 7.7×10^{140} unique masks (even for a small image; see Section 4.3); thus, it is infeasible to calculate the expected transformation. Moreover, Themis is not only randomized but also non-differentiable; in this case, it is unknown how to apply BPDA to bypassing Themis.

Adaptive AE generation. He et al. [26] describe a representative adaptive attack method against non-differentiable defences, where an attacker tries to circumvent the defensive approach by (a) considering intermediate distorted images during optimization and (b) exploring multiple diverse optimization paths.

Inspired by [26], we design similar adaptive attacks to examine the resilience of our approach. To that end, we modify the code of the CW algorithm [7], in order to adaptively generate AEs that can bypass our detector. Specifically, after each iteration in an optimization procedure, an intermediate distorted image is obtained. We then check whether it can bypass our detector. For each image, we repeat the optimization procedure for up to T times to explore different optimization paths (for this purpose, we set a randomly initialized state at the beginning of each optimization procedure). As shown in Figure 6, we set $T = 150$, corresponding to around 450 seconds on average on our machine. In comparison, the two works [50] and [26] use around 75 and 180 seconds to generate adaptive AEs for each image, respectively.

Given that adaptive CW AE generation is quite time-consuming, without loss of generality, this experiment is conducted on 500 images randomly selected from CIFAR-10. During the AE generation, we let $\kappa = 0.0$, which means that the resulting AE is classified as the target class. As κ increases, the model classifies the resulting AE as the attacker-desired label more likely. As a larger value of κ imposes an extra constraint to attackers and lowers the chance of successful adaptive attacks, we only consider $\kappa = 0.0$.

Resilience results. We adopt the SVM-based detector that achieves a detection rate of 100% (Table 1): no AEs can fool it *without adaptive attacks*. Figure shows that only 4.2% (that is, 21 AEs) of adaptive AEs can bypass our detector. By contrast, similar adaptive attacks [26] can bypass Feature Squeezing based AE detection [52] at a success rate of 100%; as another example, [50] can merely achieve a detection rate of 70% under adaptive CW attacks. More importantly, the first 50 times of the optimization path exploration attain the success rate of 3.4%, while the following 100 times only increase the success rate by 0.8%. It shows that the effect of adaptive attacks grows very slowly as the attacker doubles his time. We thus can conclude that our detection technique is not only resilient to adaptive attacks based on differentiation, but also to adaptive attacks through exploration of many optimization paths. Thus, Themis, highly resilient to adaptive CW-L_2 attacks, fills a critical gap in AE detection.

7 INTERPRETABILITY

Background. To make the final prediction, most neural-network-based image classifiers implement a *softmax* function at the last layer

$$softmax(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, \quad \text{for } i = 1, \cdots, K \text{ and } \mathbf{z} = (z_1, \cdots, z_K) \in \mathbb{R}^K \tag{3}$$

which maps an input vector $\mathbf{z}$ consisting K real numbers to a probability mass function over predicted output classes. The input vector of a *softmax* function is also called *logit*. Given a benign image whose logit is $\mathbf{z}$, the goal of an attacker is to perturb the image to get a new logit $\mathbf{z'}$ such that $\text{argmax}_i(\mathbf{z'}) \neq \text{argmax}_i(\mathbf{z})$.

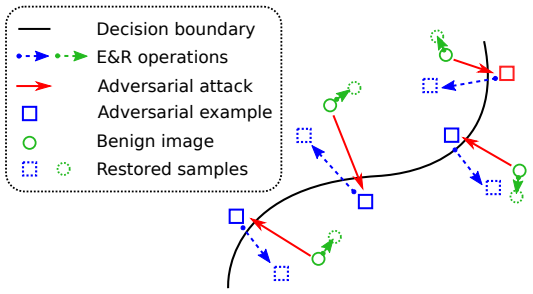


Figure 7: Illustration of how E&R works.

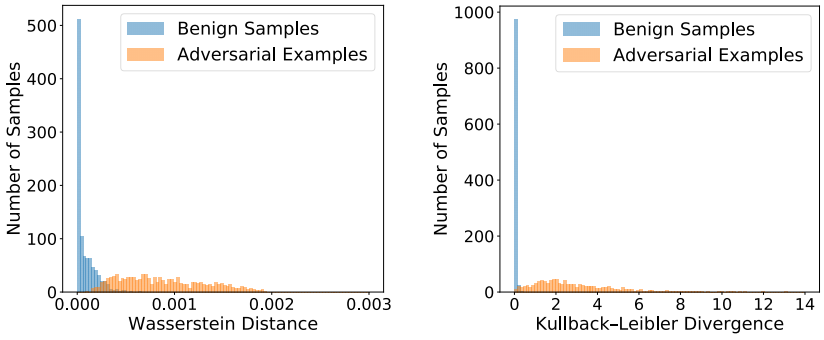
Table 6: Clusters splitting result.

Attacks	Metrics	FPR	TPR
CW-L_2	WD	0.5%	78.4%
	KL	0.0%	96.1%
DeepFool	WD	1.1%	85.7%
	KL	0.5%	89.3%

Interpretation Using Classification Results. Let $f(x)$ be the output of the *softmax* layer of a neural network f when feeding the input x. Let $T(x)$ be the output of processing x with E&R operations. If x is benign, since it is not sensitive to E&R operations, the probability mass functions $f(x)$ and $f(T(x))$ are similar. By contrast, if x is an AE, $f(x)$ is significantly different from $f(T(x))$, since AEs are very sensitive to E&R operations. In short, if the sensitivity distinction between AEs and benign samples is true, the divergence (or distance) between $f(x)$ and $f(T(x))$ should reflect whether x is malicious or benign. We then adopt two widely used metrics, Wasserstein distance (WD for short) [51] and Kullback-Leibler divergence (KL for short) [29].

As shown in Figure 7, we depict benign and adversarial examples by green circles and blue squares, respectively. The arrows with dotted line represent E&R operations. We consider the changes caused by E&R operations on benign images and AEs (depicted by green and blue arrows with dotted line, respectively) should fall into different probability distributions. To visualize this, we randomly select 1,000 image pairs consisting of AEs and benign instances from $\mathcal{D}_I$-Test. After feeding them (with and without applying E&R operations) into the image classification model, we collect the output of the *softmax* layer. Then, we measure the difference between $f(x)$ and $f(T(x))$. To be consistent with the design of Themis, we apply E&R operations 10 times for each image and calculate an arithmetic mean of the 10 measurements. The visualization of samples is shown in Figure 8, which confirms our proposition; that is, the changes caused by E&R operations on benign images and AEs fall into different clusters.

Next, we quantitatively analyse to what extent the distance/divergence measurement can help discriminate an AE that is across the decision boundary. In detail, we use an optimal threshold based on the ROC (receiver operating characteristic) curve, to split AEs and benign images distributions. Table 6 presents the FPR and TPR (i.e., Detection Rate defined in Section 5). Note that the results are



(a) CW-L_2 (WD) (b) CW-L_2 (KL)

(c) DeepFool (WD) (d) DeepFool (KL)

Figure 8: Visualization of the changes caused by E&R on benign samples and AEs.

only for illustrating that E&R imposes different impacts on AEs and benign samples in terms of probability mass function changes, and do not represent the detection performance of Themis (see Section 5 for its detection performance). Here, we only use one dimensional feature (i.e., the Wasserstein distance or KL divergence) to split two clusters, information loss inevitably degrades the splitting performance, which is mitigated by the design of Themis.

Interpretation through Visualization of Feature Vectors. The feature vectors due to 1,000 randomly selected benign samples from the ImageNet dataset and the corresponding 1,000 AEs are visualized in Figure 9. For the visualization purpose, it shows only three principal components of the pre-processed feature vectors (see Figure 4). We have two observations. (1) While the feature vectors of benign samples, before and after the E&R operations, are close (Figure 9(a)), those of AEs form two clusters far apart (Figure 9(b)). (2) PCA is effective in preserving features that help distinguish benign samples from AEs.

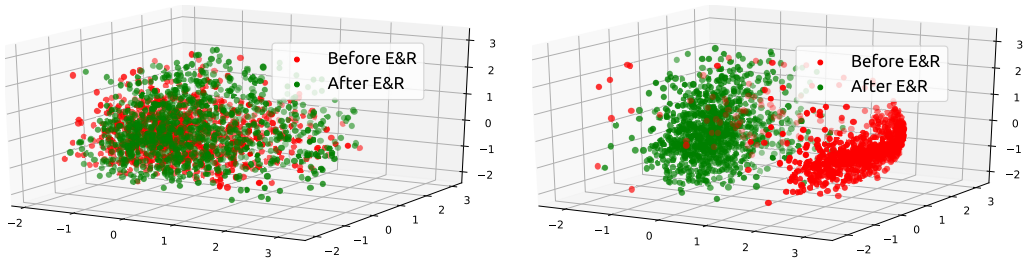
8 RELATED WORK

Countermeasures against AE attacks can be roughly divided into two categories. The first category aims to eliminate the influences of AEs by either rectifying them or fortifying the target neural network itself. The second category is AE detectors (including our work), the goal of which is to predict whether an input is adversarial, so that the target neural network can reject those inputs. Given the large body of research on AEs, this is not intended to be exhaustive.

8.1 Adversarial Influences Elimination

To improve the robustness of neural networks, *adversarial training* augments the training set with the label-corrected AEs [35, 54]. Buckman et al. [4] propose using thermometer-encoded inputs to assist adversarial training. Alternatively, *Shield* [13] enhances a model by re-training it with multiple levels of compressed images using JPEG, a commonly used image compression technique.

Another strategy is to pre-process the inputs before feeding them to neural networks. For instance, the pixel deflection and a wavelet-based denoiser are combined to rectify AEs [43]. Liao et al. [32] propose higher-level guided denoisers aiming to remove the adversarial noise from inputs. Some other methods adopt JPEG compression techniques [23, 44] to filter out the information redundancy, which otherwise provides living space for adversarial perturbations. However, their accuracies under adaptive attacks are lack of adequate evaluations. CIIDefence [24] proposes to use image inpainting with wavelet based denoising to rectify the classification result. However, its inpainting mask is guided by class activation maps, which can be predicted and exploited by an adaptive attacker. Both MagNet [37] and [8] essentially take the path of removing noises/enhancing images, rather than the Erase-and-Restore path proposed in this work. REMIX [8] applies inpainting to rectifying



(a) Benign samples (b) AEs

Figure 9: Visualization of feature vectors. The coordinate axes respectively represent three largest principal components.

classification results, with an rectifying accuracy 86% on CIFAR-10. It uses autoencoder as the inpainter. Autoencoders are typically data-specific, which means that it is only effective on images similar to what they have been trained on. It did not study the resilient to adaptive attacks and did not provide interpretation either.

Unlike all these works, the purpose of our work is for highly accurate attack detection, e.g., an accuracy of over 98% on CIFAR-10 and ImageNet. It does not have dependency on high similarity between training data and testing data. It is target-model agnostic: a detector trained using AEs targeting one model can be directly used to detect AEs targeting another. Moreover, our work provides interpretation why the detection method works, and carefully examines its resilience to adaptive attacks.

8.2 Adversarial Examples Detection

Li et al. [31] extract PCA features after inner convolutional layers of the DNN, and then use a cascade classifier to detect AEs. Metzen et al. [38] train a CNN-based auxiliary network. This light-weight sub-network works with the target model to detect AEs. Some techniques apply pre-processors on input images and use prediction mismatch strategy to detect AEs. For example, Meng et al. [37] train an auto-encoder as the image filter. If the predictions of an original image and the corresponding processed one fail to match, the input is adversarial. Similarly, Xu et al. [52] propose Feature Squeezing to detect AEs by comparing the prediction for the original input with that for the squeezed one. However, adaptive attacks have successfully circumvented all of the aforementioned detection methods [5, 6, 26]. Finally, Tian et al. [50] leverage image rotation and shifting as pre-processors to construct a detector. Although these operations can produce certain randomness to counter some adaptive attacks, their randomness pool is very limited. It only has 45 possible transformations. As a result, their method can merely achieve a detection rate of 70% under adaptive attacks [50].

Zeng et al. [53] proposes a novel AE detection method inspired by multiversion programming, which first uses multiple off-the-shelf audio recognition systems to classify the same audio input and then compares the classification results to detect AEs. Their insight is the extraordinary difficulty of generating highly transferable audio AEs, which is not the case for image AEs. We also make use of multiple classification results, which, however, is based on the idea of sampling (i.e., applying E&R multiple times) to enhance the detection accuracy.

Table 7: Performance of integrating Themis with an existing detector [55].

Classifier	FPR	Detection Rate			
		CW-L_0	JSMA	CW-L_2	DeepFool
SVM	3.4%	98.8%	99.6%	97.2%	98.0%
AdaBoost	1.5%	98.8%	99.6%	96.4%	97.2%

To our knowledge, our prior work [55] is the first that proposes to use inpainting for AE detection, but it applies inpainting in a different way from this work. Specifically, [55] focuses on detecting L_0 attacks by inpainting salient noises, as L_0 attacks usually cause large-amplitude perturbations due to minimizing the number of modified pixels.

The AE detection idea that *intentionally* and *randomly* "damages" (i.e., erases) some pixels of an image and then uses an inpainting algorithm is not only ingenious and effective, but can also be interpreted and keep resilient to adaptive attacks. Unlike other very complex methods, our method is extremely simple and easy to apply. As discussed in Section 9, although it only handles L_2 attacks, it can easily work as a plugin or complement to enhance an existing attack detection system.

9 DISCUSSION AND FUTURE WORK

While our work focuses on detecting L_2 AEs, it is easy to combine our approach with other detectors that show strengths in detecting other types of AEs to build a comprehensive hybrid detector. A simplest integration is that *an input is detected as an AE if any of the integrated detectors reports so.* To illustrate this, as an example, we integrate Themis with our detection system [55] specialized in detecting L_0 attacks to build a more *comprehensive* detector. Table 7 shows the performance of this hybrid detector.

The proposed erasing and restoring approach works by destruction of the carefully perturbed pixels. Attackers thus may consider minimizing the number of perturbed pixels, like in L_0 AEs, to evade our detection. However, the prior work points out that L_0 AE generation results in large amplitudes of altered pixels, which can be exploited to locate and restore most of the maliciously perturbed pixels [55]. Therefore, for the purpose of AE generation, making a trade-off between the number of altered pixels and their resulting amplitudes is a direction worth exploration.

Another possible adaptive attack is to limit the perturbations in a restricted area that the defender is not aware of. Most prior works [30, 39, 47] that limit perturbed pixels to a given sub-region use L_0-norm. We notice that some recent works [14, 17] that only perturb pixels in a limited region also use L_2-norm to achieve better invisibility. However, their modified regions or even pixels are predictable, which can be exploited by an AE detector. Therefore, how to limit the L_2 perturbation to an *arbitrary* sub-region is still an open question. A future task is to investigate the effectiveness of E&R once such L_2 perturbations are available.

This work focuses on attacks launched against digital images; we notice that physical attacks [18, 19] are attracting more and more interests from the research community. In particular, patch-based AEs, which are widely used in physical attacks, are not in the scope of this work. However, it is interesting to study the effectiveness of E&R on physical attacks [19]. We leave this as our future work.

Finally, some recent studies on certified robustness have attracted much interest from the research community. For example, Cohen et al. [10] present a certified robustness guarantee in L_2 norm for the smoothed classifier that is obtained by using Gaussian noise. Furthermore, Jia et al. [27] derive a tight robustness in L_2 norm for top-k predictions when using randomized smoothing with Gaussian noise. Some related works [1, 46] also show that inpainting has a side effect of denoising by smoothing the interpolated pixels. Our E&R approach can be considered as an alternative to randomized smoothing. Thus, it is interesting to analyze the certified accuracy of our E&R method. We plan to explore this in our future work.

10 CONCLUSION

Our finding has revealed that L_2 AEs are sensitive to the Erase-and-Restore operations, while benign samples are not. Exploiting the sensitivity distinction, we have proposed a novel and effective AE detection approach E&R. It outperforms other state-of-the-art approaches in terms of both detection rates and false positive rates. In addition, our detector is target-model agnostic, keeps effective across different L_2 attack methods (i.e., good transferability across attack methods), and is resilient to adaptive attacks. Furthermore, we have interpreted the detection technique from both qualitative and quantitative angles to provide deeper understanding of the technique. Unlike many other detection methods that are complex and thus difficult to construct and train, this method is very simple to build and easy to apply in practice.

ACKNOWLEDGMENTS

We would like to thank our shepherd, Dr. Qi Alfred Chen, and the anonymous reviewers for their invaluable suggestions. This work was supported in part by the US National Science Foundation (NSF) under grants CNS-1856380 and CNS-2016415.

REFERENCES

[1] Robin Dirk Adam, Pascal Peter, and Joachim Weickert. 2017. Denoising by inpainting. In *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer, 121–132.

[2] Anish Athalye, Nicholas Carlini, and David Wagner. 2018. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*.

[3] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. 2018. Synthesizing Robust Adversarial Examples. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*.

[4] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. 2018. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations (ICLR)*.

[5] Nicholas Carlini and David Wagner. 2017. Adversarial examples are not easily detected: Bypassing ten detection methods. In *ACM Workshop on Artificial Intelligence and Security*.

[6] Nicholas Carlini and David Wagner. 2017. Magnet and "efficient defenses against adversarial attacks" are not robust to adversarial examples. *arXiv preprint arXiv:1711.08478* (2017).

[7] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy (SP)*.

[8] Kang-Cheng Chen, Pin-Yu Chen, and Chia-Mu Yu. 2018. Poster: REMIX: Mitigating Adversarial Perturbation by Reforming, Masking and Inpainting. In *IEEE Symposium on Security and Privacy (SP)*.

[9] François Chollet. 2015. Keras. https://keras.io.

[10] Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. 2019. Certified adversarial robustness via randomized smoothing. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*.

[11] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995).

[12] Gabriel B Paranhos da Costa, Welinton A Contato, Tiago S Nazare, João ES Neto, and Moacir Ponti. 2016. An empirical study on the effects of different types of noise in image classification tasks. *arXiv preprint arXiv:1609.02781* (2016).

[13] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, Li Chen, Michael E Kounavis, and Duen Horng Chau. 2018. Shield: Fast, practical defense and vaccination for deep learning using JPEG compression. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

[14] Ting Deng and Zhigang Zeng. 2019. Generate adversarial examples by spatially perturbing on the meaningful area. *Pattern Recognition Letters* 125 (2019), 632–638.

[15] Steven Diamond, Vincent Sitzmann, Stephen Boyd, Gordon Wetzstein, and Felix Heide. 2017. Dirty pixels: Optimizing image classification architectures for raw sensor data. *arXiv preprint arXiv:1701.06487* (2017).

[16] Samuel Dodge and Lina Karam. 2016. Understanding how image quality affects deep neural networks. In *IEEE International Conference on Quality of Multimedia Experience (QoMEX)*.

[17] Xiaoyi Dong, Dongdong Chen, Jianmin Bao, Chuan Qin, Lu Yuan, Weiming Zhang, Nenghai Yu, and Dong Chen. 2020. GreedyFool: Distortion-Aware Sparse Adversarial Attack. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[18] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, Tadayoshi Kohno, and Dawn Song. 2018. Physical adversarial examples for object detectors. In *12th USENIX Workshop on Offensive Technologies (WOOT)*.

[19] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. 2018. Robust physical-world attacks on deep learning visual classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[20] Yoav Freund and Robert E Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences* 55, 1 (1997).

[21] Zhitao Gong, Wenlu Wang, and Wei-Shinn Ku. 2017. Adversarial and clean data are not twins. *arXiv preprint arXiv:1704.04960* (2017).

[22] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. 2017. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280* (2017).

[23] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. 2018. Countering adversarial images using input transformations. In *International Conference on Learning Representations (ICLR)*.

[24] Puneet Gupta and Esa Rahtu. 2019. CIIDefence: Defeating Adversarial Attacks by Fusing Class-Specific Image Inpainting and Image Denoising. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[26] Warren He, James Wei, Xinyun Chen, Nicholas Carlini, and Dawn Song. 2017. Adversarial example defenses: Ensembles of weak defenses are not strong. In *11th USENIX Workshop on Offensive Technologies (WOOT)*.

[27] Jinyuan Jia, Xiaoyu Cao, Binghui Wang, and Neil Zhenqiang Gong. 2020. Certified robustness for top-k predictions against adversarial perturbations via randomized smoothing. In *International Conference on Learning Representations (ICLR)*.

[28] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

[29] Solomon Kullback. 1997. *Information theory and statistics*. Courier Corporation.

[30] Hyun Kwon, Hyunsoo Yoon, and Daeseon Choi. 2019. Restricted evasion attack: Generation of restricted-area adversarial example. *IEEE Access* 7 (2019), 60908–60919.

[31] Xin Li and Fuxin Li. 2017. Adversarial examples detection in deep networks with convolutional filter statistics. In *Proceedings of the IEEE International Conference*

on Computer Vision (ICCV).

[32] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Jun Zhu, and Xiaolin Hu. 2018. Defense against adversarial attacks using high-level representation guided denoiser. In *Conference on Computer Vision and Pattern Recognition (CVPR).*

[33] Shiqing Ma, Yingqi Liu, Guanhong Tao, Wen-Chuan Lee, and Xiangyu Zhang. 2019. NIC: Detecting Adversarial Samples with Neural Network Invariant Checking. In *Network and Distributed System Security Symposium (NDSS).*

[34] Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. 2018. Characterizing adversarial subspaces using local intrinsic dimensionality. In *International Conference on Learning Representations (ICLR).*

[35] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR).*

[36] Julien Mairal, Michael Elad, and Guillermo Sapiro. 2007. Sparse representation for color image restoration. *IEEE Transactions on image processing* 17, 1 (2007).

[37] Dongyu Meng and Hao Chen. 2017. MagNet: a two-pronged defense against adversarial examples. In *ACM SIGSAC Conference on Computer and Communications Security (CCS).*

[38] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. 2017. On detecting adversarial perturbations. In *International Conference on Learning Representations (ICLR).*

[39] Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. 2019. Sparsefool: a few pixels make a big difference. In *Conference on Computer Vision and Pattern Recognition (CVPR).*

[40] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. DeepFool: a simple and accurate method to fool deep neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR).*

[41] Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, Ian Molloy, and Ben Edwards. 2018. Adversarial Robustness Toolbox v1.0.1. https://adversarial-robustness-toolbox.readthedocs.io/en/latest/modules/attacks/evasion.html#carlini-and-wagner-l-2-attack.

[42] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016. Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE Symposium on Security and Privacy (SP).*

[43] Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. 2018. Deflecting adversarial attacks with pixel deflection. In *Conference on Computer Vision and Pattern Recognition (CVPR).*

[44] Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. 2018. Protecting JPEG images against adversarial attacks. In *2018 Data Compression Conference.* IEEE, 137–146.

[45] Jonas Rauber, Wieland Brendel, and Matthias Bethge. 2017. Foolbox: A Python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131* (2017).

[46] Toby Sanders and Christian Dwyer. 2019. Inpainting versus denoising for dose reduction in scanning-beam microscopies. *IEEE Transactions on Image Processing* 29 (2019), 351–359.

[47] Ali Shafahi, W Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. 2019. Are adversarial examples inevitable?. In *International Conference on Learning Representations (ICLR).*

[48] Jianhong Shen and Tony F Chan. 2002. Mathematical models for local nontexture inpaintings. *SIAM J. Appl. Math.* 62, 3 (2002).

[49] Alexandru Telea. 2004. An image inpainting technique based on the fast marching method. *Journal of Graphics Tools* 9, 1 (2004).

[50] Shixin Tian, Guolei Yang, and Ying Cai. 2018. Detecting Adversarial Examples through Image Transformation. In *AAAI Conference on Artificial Intelligence.*

[51] Cédric Villani. 2009. The Wasserstein distances. In *Optimal Transport.* Springer, 93–111.

[52] Weilin Xu, David Evans, and Yanjun Qi. 2018. Feature squeezing: Detecting adversarial examples in deep neural networks. In *Network and Distributed System Security Symposium (NDSS).*

[53] Qiang Zeng, Jianhai Su, Chenglong Fu, Golam Kayas, Lannan Luo, Xiaojiang Du, Chiu C Tan, and Jie Wu. 2019. A multiversion programming inspired approach to detecting audio adversarial examples. In *49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN).*

[54] Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. 2016. Improving the robustness of deep neural networks via stability training. In *Conference on Computer Vision and Pattern Recognition (CVPR).*

[55] Fei Zuo, Bokai Yang, Xiaopeng Li, Lannan Luo, and Qiang Zeng. 2019. Exploiting the Inherent Limitation of L_0 Adversarial Examples. In *International Symposium on Research in Attacks, Intrusions and Defenses (RAID).*