Detecting Adversarial Image Examples in Deep Neural Networks with Adaptive Noise Reduction

Bin Liang, Hongcheng Li, Miaoqiang Su, Xirong Li, Wenchang Shi and Xiaofeng Wang

Abstract—Recently, many studies have demonstrated deep neural network (DNN) classifiers can be fooled by the adversarial example, which is crafted via introducing some perturbations into an original sample. Accordingly, some powerful defense techniques were proposed. However, existing defense techniques often require modifying the target model or depend on the prior knowledge of attacks. In this paper, we propose a straightforward method for detecting adversarial image examples, which can be directly deployed into unmodified off-the-shelf DNN models. We consider the perturbation to images as a kind of noise and introduce two classic image processing techniques, *scalar quantization* and *smoothing spatial filter*, to reduce its effect. The image entropy is employed as a metric to implement an adaptive noise reduction for different kinds of images. Consequently, the adversarial example can be effectively detected by comparing the classification results of a given sample and its denoised version, without referring to any prior knowledge of attacks. More than 20,000 adversarial examples against some state-of-the-art DNN models are used to evaluate the proposed method, which are crafted with different attack techniques. The experiments show that our detection method can achieve a high overall F1 score of 96.39% and certainly raises the bar for defense-aware attacks.

Index Terms—Adversarial example, deep neural network, detection

1 Introduction

Deep neural networks (DNNs) [1] have been widely adopted in many applications such as computer vision [2], [3], speech recognition [4], [5], and natural language processing [6], [7]. DNNs have exhibited very impressive performance in these tasks, especially in the image classification [3]. Some DNN-based classifiers achieved even better performance than human [8], [9]. Meanwhile, their robustness has also raised concerns.

Some recent studies [10], [11], [12] demonstrate that DNN-based image classifiers can be fooled by *adversarial examples*, which are well-crafted to cause a trained model to misclassify. For example, as presented in [10], added with some imperceptible perturbations, an image can be misclassified with very high confidence by GoogLeNet [3], while a human observer can still correctly classify it without noticing the existence of the introduced perturbations. These studies indicate that the adversaries could potentially use the crafted image to inflict serious damages. As shown in [13], a *stop* sign, after being crafted, will be incorrectly classified as a *yield* sign. As a result, a self-driving car equipped with the DNN classifier may behave dangerously.

Some techniques have been proposed to defend adversarial examples in DNNs [10], [14], [15], [16]. Most of them require modifying the target model. For example, the *adversarial training* is a straightforward defense technique which uses as many adversarial examples as possible during training process as a kind of regularization [10], [14], [15]. Papernot et al. [16] introduced a defense technique named *defensive distillation*. Two networks were trained as a distillation, where the first network produced probability vectors to label the original dataset, while the other was trained using the newly labeled dataset. As a result, the effectiveness of adversarial examples can be substantially reduced. Several very recent studies [17], [18], [19], [20] focus on detecting adversarial examples directly. Similarly, these techniques also require modifying the model or acquiring sufficient adversarial examples, such as training new sub-models [17], retraining a revised model as a detector using known adversarial examples [18], performing a statistical test on a large group of adversarial and benign examples [19], or training the key detection parameter using a number of adversarial examples and their corresponding benign ones [20].

Unfortunately, retraining an existing model or changing its architecture will introduce expensive training cost. Generating appropriate adversarial examples for training or statistical testing is also of high cost and depends on a very comprehensive prior knowledge of various potential adversarial techniques. Even worse, the attacker can craft adversarial examples with the technique unknown to the defender. In this case, the adversarial example has a good chance to evade the classifier. Moreover, training a classifier with an emerging attack technique would take some time. There always is a window for attackers to craft effectual adversarial examples. Furthermore, most of existing defense techniques are model-specific. To apply a defense technique to different models, they need to be rebuilt or retrained individually. The security enhancement to a model cannot be directly applied to other ones.

To address the aforementioned challenges, we present in this paper a new technique capable of effectively capturing adversarial examples, even in the absence of prior knowledge about potential attacks.

Our approach is based upon the observation that to

- *B. Liang, H. Li, M. Su, X. Li and W. Shi are with the School of Information, Renmin University of China, Beijing, China; and also with Key Laboratory of DEKE (Renmin University of China), MOE, China. Email: {liangb, owenlee, sumiaoqiang, xirong, wenchang}@ruc.edu.cn.*
- *X. Wang is with the Center for Security Informatics, Indiana University Bloomington, Bloomington, IN 47405 USA. Email: xw7@indiana.edu.*

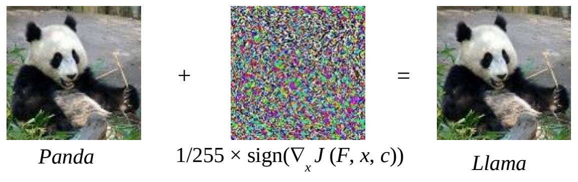


Fig. 1: Very small perturbation can result in an effective adversarial example for a color image.

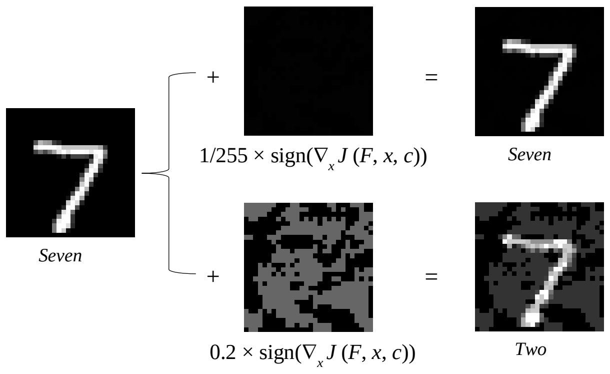


Fig. 2: Only large perturbation can produce an adversarial example for a simple image.

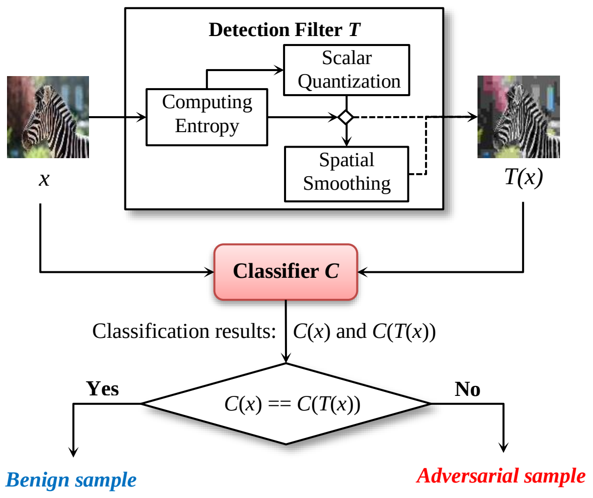


Fig. 3: Detection Framework.

make the adversarial change imperceptible, the perturbation incurred by the adversarial examples typically needs to be confined within a small range. This is important, since otherwise, the example will be easily identified by human. Consequently, the information introduced by the perturbation should also be less than that of the original image. In the proposed method, we treat the perturbation as a kind of artificial noise and leverage the noise reduction techniques to reduce its adversarial effect. If the effect is downgraded properly, the denoised adversarial example will be classified as a new class that is different from the adversarial target. On the other hand, for the legitimate sample, the same denoising operation will most likely just slightly changes the image's semantics, keeping it still within its original category. Intuitively, all the adversarial perturbation is added later on to the image and therefore tends to less tolerant of the noise reduction process than the original image information. The information remaining in a denoised benign sample can be still enough for the classifier to correctly identify its class. In fact, some studies [21], [22] have shown that the state-of-the-art classifiers are somewhat robust against certain degree of distortions. To this end, the adversarial example can be effectively detected by inspecting whether the classification of a sample is changed after it is denoised.

Two classic image processing techniques, *scalar quantization* and *smoothing spatial filter*, are leveraged to reduce the effect of perturbations. However, it is actually inappropriate to denoise all samples in the same way. A simple image, e.g., a grayscale handwritten digit, may only consist of hundreds of pixels. In other words, the perturbation space is limited when generating an adversarial example for this kind of image. The adversary have to magnify the perturbation to generate an effective adversarial example [10]. On the other hand, a color photo can often provide a larger pertubation space. Adding a small scale pertubation can make it be misclassified. As Fig. 1 illustrates, manipulating a color image with very small perturbation can generate an effective adversarial example with the attack method in [10]. However, using the same strength perturbation fails to generate adversarial example for a handwriting digit, as Fig. 2 shows. Only when we magnify the perturbation to at least 50 times, can we get an effective adversarial example. As a result, the quantization or smoothing suitable for a high-resolution sample may not be enough for a low-resolution one. Using a uniform noise reduction strategy may over-denoise some samples or under-denoise some others, and introduce false positives and negatives. Obviously, we should employ different denoising strategies for different images.

To improve the generality of our method for different kinds of images, an adaptive noise reduction is enforced. The *entropy* of the image is utilized to measure the complexity of samples. An aggressive noise reduction strategy is adopted for the sample with a low entropy, which may be perturbed heavily by the adversary. For high-entropy samples, a light strategy is employed. Specifically, as illustrated in Fig. 3, the key component of our method is a detection filter T. When feeding a sample x to the target classifier C, it will be denoised by T to generate a filtered sample $T(x)$. The sample is first quantized with an appropriate interval size, which is determined by computing the entropy of the sample. We also use the entropy to decide whether the quantized sample needs to be smoothed. Only when the entropy is larger than a threshold, will it be smoothed by a spatial smoothing filter. Finally, two individual classifications are performed. The one is for the original sample x, and the other is for the denoised sample $T(x)$. Only when the two classifications are equal, i.e., $C(x) == C(T(x))$, will x be considered benign. Otherwise, it is identified adversarial. In practice, the detection mechanism cannot know the true class of x in advance to validate its current classification $C(x)$. In our method, $C(T(x))$ actually acts as a detection baseline to detect potential adversarial examples.

We employ some state-of-the-art DNN models and popular datasets, such as GoogLeNet [3], CaffeNet [23], MNIST [24] and ImageNet [25] to evaluate the effectiveness of the proposed method. Three up-to-date attack techniques, i.e., FGSM [10], DeepFool [11], and CW attacks [26], are used to craft adversarial examples. We evaluate our method in two attack scenarios, namely, *defense-unaware* and *defense-aware*. In the defense-unaware scenario, we generated 21,673 effectual adversarial examples in total. The experiments show that the proposed method can achieve an overall recall of 95.00% and an overall precision of 97.81% for detecting the adversarial examples, resulting in a high overall F1 score [27] of 96.39%. Compared with other recent detection methods, our method can detect more adversarial examples with fewer false positives. In the defense-aware scenario, the experiments show that our detection method can effectively raise the bar for adversaries, the attack success rate is remarkably downgraded to 67.37%. As demonstrated in [28], many recent detection methods are defeated by defense-aware attacks. In contrast, the proposed method is proven to be a promising mitigation to defense-aware attacks.

In summary, our main contributions are the following.

- We model the perturbation of the DNN adversarial examples as image noise and introduce classic image processing techniques to reduce its effect. This allow us to effectively detect adversarial examples without prior knowledge of attack techniques.
- We employ the entropy to automatically adjust the detection strategy for a specific sample. This makes the proposed method capable of detecting different kinds of adversarial examples without requiring tuning its parameters, and can be directly integrated into unmodified target models.
- Using some state-of-the-art DNN models, we demonstrate that the proposed method can effectively detect the adversarial examples generated by different attack techniques with a better performance than other recent detection method, especially in the defense-aware attack scenario.[1]

2 Background

In this section, we provide some preliminaries on attack scenarios and the attack techniques used in our experiments.

2.1 Attack Scenarios

In practice, the adversaries may possess different levels of knowledge of target models. Accordingly, they may adopt different methodologies when launching attacks. Similarly as done in [20], [28], [29], we evaluate our detection method in two different attack scenarios, i.e., *defense-unaware* and *defense-aware*.

Defense-unaware Scenario. As discussed in [30], more and more well-trained DNN models can be obtained from online markets (e.g., Caffe Model Zoo). Some users may directly employ such public models in their tasks [31], [32]. For example, Esteva et al. [31] utilized a pre-trained Inception v3 model and retrained it on some clinical images to classify skin cancers. Both the original model and the retraining samples are publicly available. In other words, the adversary can also directly get the target model or build a same one and thoroughly analyze it. However, as illustrated in Fig. 3, the proposed technique can be easily deployed on an unchanged trained model. The key component of the proposed technique is a filter T, which can be directly integrated with an original model C without requiring any modification to it. In fact, the user can stealthily introduce our detection technique in a public model. In this scenario, the adversary can completely understand C but has no idea about the existence of T (i.e., *defense-unaware*).

Defense-aware Scenario. In some extreme cases, the adversaries can by all means get the whole target model, including the integrated detection mechanism. Consequently, they can fully analyze both C and T to generate desired adversarial examples and launch more sophisticated attacks.

2.2 Crafting Adversarial Example

Szegedy et al. [12] first made the intriguing discovery that various machine learning models, including DNNs [33], are vulnerable to adversarial examples. In general, for a given sample x and a trained model C, the attacker aims to craft an adversarial example $x^* = x + \Delta x$ by adding a perturbation Δx to x, such that $C(x^*) \neq C(x)$.

In most of the cases, the attacker wants the target model to misclassify the resultant image, while a human observer can still correctly classify it without noticing the existence of the introduced perturbation. In practice, the adversarial examples can be generated straightforwardly [10] or with an optimization procedure [11], [12], [26]. In this paper, we choose the following three up-to-date attack techniques to perform detection experiments. They can produce imperceptible perturbations.

Fast Gradient Sign Method. Goodfellow et al. [10] proposed a straightforward strategy named fast gradient sign method (FGSM) to craft adversarial examples against GoogLeNet [3]. The method is easy to implement and can compute adversarial perturbations very efficiently. Let c be the true class of x and $J(C, x, c)$ be the cost function used to train the DNN C. The perturbation is computed as the sign of the model's cost function gradient, i.e.

$$\Delta x = \varepsilon \ sign(\bigtriangledown_X J\left(C,\ x,\ c\right)) \tag{1}$$

where ε (ranging from 0.0 to 1.0) is set to be small enough to make Δx undetectable. Choosing a small ε can produce a well-disguised adversarial example. The change to the original image is difficult to be spotted by a human. On the contrary, a large ε is likely to introduce noticeable perturbations but can get more adversarial examples when the original images are simple (e.g., handwritten digits).

In the classic FGSM algorithm, all input pixels are applied either a positive or negative change in the same degree according to the direction (sign) of corresponding cost gradients. However, as illustrated in Fig. 4, we found that only manipulating the 30,000 (19.92%) input pixels with the highest positive or negative gradient magnitude can also generate an effectual adversarial example using the same ε. The result implies that we can't assume the perturbation follows some kind of distribution.

1. The source code of our detection method, along with the experiment data, is all available at https://github.com/OwenSec/DeepDetector.

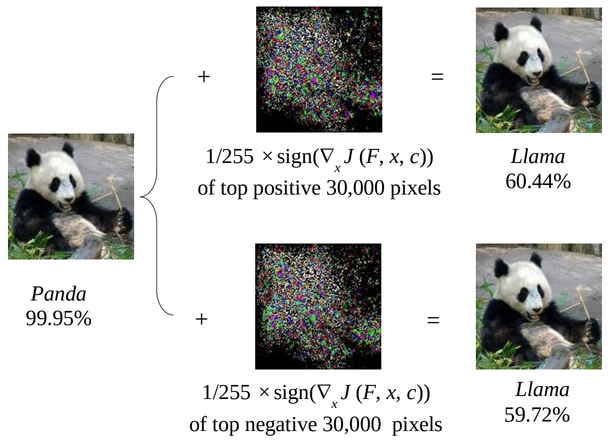

Panda
99.95%
+
=
1/255 × sign($\nabla_x J(F, x, c)$)
of top positive 30,000 pixels
Llama
60.44%
+
=
1/255 × sign($\nabla_x J(F, x, c)$)
of top negative 30,000 pixels
Llama
59.72%

Fig. 4: Only manipulating the pixels with top gradients can still result in effectual adversarial examples.

DeepFool. Moosavi-Dezfooli et al. [11] devised the DeepFool algorithm to find very small perturbations that are sufficient to change the classification result. The original image x_0 is manipulated iteratively. At each iteration of the algorithm, the perturbation vector for x_i that reaches the decision boundary is computed, and the current estimate is updated. The algorithm stops until the predicted class of x_i changes. DeepFool is implemented as an optimization procedure which can yield a good approximation of the minimal perturbation. Moosavi-Dezfooli et al. performed some attack experiments against several DNN image classifiers, such as GoogLeNet [3] and CaffeNet [23], and so on. Their experiments demonstrated that DeepFool can lead to a smaller perturbation than FGSM, which however is still effective to trick the target models.

CW Attacks. Carlini and Wagner [26] also employed an optimization algorithm to seek as small as possible perturbations. Three powerful attacks (CW attacks for short) are designed for the L_0, L_2, and L_∞ distance metrics. Using some public datasets, such as MNIST [24] and ImageNet [25], they trained some deep network models to evaluate their attack methods. As demonstrated in [26], CW attacks can find closer adversarial examples than the other attack techniques and never fail to find an adversarial example. For example, CW L_0 and L_2 attacks can find adversarial examples with at least 2 times lower distortion than FGSM. Besides, Carlini and Wagner also illustrated their attacks can effectively break the defensive distillation [16].

3 Methodology

3.1 Overview

The basic idea behind our method is to regard the perturbation as a kind of noise and introduce image processing techniques to reduce its adversarial effect as far as possible.

As described in Section 2, an adversarial example is crafted by superimposing some perturbations on the original image. In this sense, the perturbation introduced in the adversarial example is an additive noise item $\eta(m, n)$, and the adversarial example can be considered as a degraded image $g(m, n)$ of the original image $f(m, n)$ as follows,

$$g(m,n) = f(m,n) + \eta(m,n) \tag{2}$$

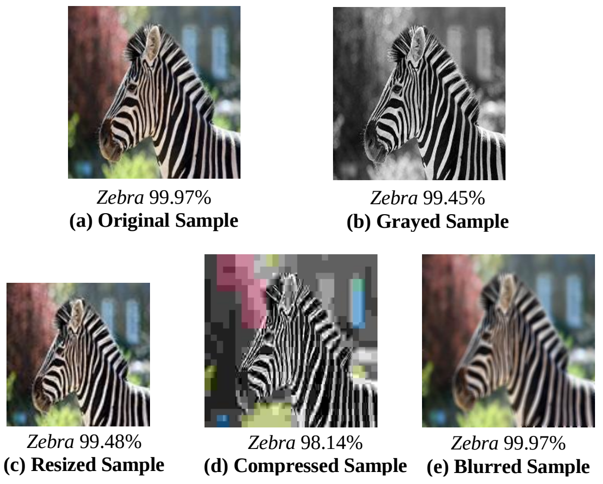


Fig. 5: GoogLeNet can correctly classify the processed images.

where m and n are spatial coordinates, and f, g and η are the functions mapping a pixel of coordinate (m, n) to its intensity.

For example, the perturbation of an FGSM adversarial example can be viewed as an additive noise whose amplitude is ε. Ideally, if we can reconstruct the original image $f(m, n)$ from an adversarial example $g(m, n)$, adversarial examples can be detected immediately. However, achieving this is very difficult, if not impossible, due to lack of the necessary knowledge about the noise term $\eta(m, n)$. Instead, we seek to reconstruct the original image in the sense of classification. Namely, we want to convert $g(m, n)$ to a new image $f'(m, n)$ such that its predicted class $C(f'(m, n))$ can be the same as $C(f(m, n))$.

Naturally, we hope that the model can correctly identify a benign sample after the conversion. If so, the adversarial example can be detected by checking whether the classification of a sample is changed. If the classification is changed, the sample will be identified as a potential adversarial example. Otherwise, it is considered benign. Fortunately, the state-of-the-art DNN image classifiers are robust to some extent and can tolerate a certain degree of distortion [21], [22], although they are weak when facing adversarial examples. As Fig. 5 illustrates, the original image is classified as *Zebra* by GoogLeNet with 99.97% confidence. After being either *grayed*, *resized*, *compressed* or *blurred*, it will still be correctly classified with high confidence.

As mentioned in Section 1, the noise reduction techniques are leveraged to reduce the effect of the perturbation. Based on the above discussions, we believe that although some details of interest in the sample may be removed too, the classifiers can output the correct classification for a denoised image. Note that, though the perturbation is treated as a kind of noise in this study, we cannot imply that the perturbation follows certain distributions. As presented in [11], the perturbation may differ largely from one attack to another. This makes identifying them a nontrivial work. Moreover, in real-life scenarios, we are unlikely to know what technique the adversary uses and what features the perturbation may possess. In fact, the introduced perturbation can be random and there is not a predictable distribution about them, as illustrated in Fig. 4. For this reason, rather than the prior knowledge-based techniques (e.g. *Lee filtering* [34]), we adopt two straightforward techniques that require no prior knowledge, namely *scalar quantization* and *smoothing spatial filter*, to detect adversarial examples.

For scalar quantization, the size of intervals is a key parameter. In principle, using large intervals can more effectively reduce the effect of the perturbation but introduce more distortions at the same time, and the "business" of an image is damaged more heavily. This may result in a misclassification for a quantized benign sample, and produce a false positive. On the contrary, smaller intervals may bring more false negatives due to inadequate noise reduction.

We utilize the image *entropy* to determine the parameter. The concept of image entropy describes how much randomness (or uncertainty) an image possesses, in other words, how much information is provided by the image [35]. The entropy value of an image is a quantitative measure of the information transmitted by the image. Commonly, the higher the entropy of an image is, the richer its semantics ("business") often is. Consequently, for an image with higher entropy, more information is required for the classifier to identify its class. Based on the intuition, to avoid excessively eliminating the information of a sample, a small interval size will be applied to the high-entropy samples when quantizing them. Accordingly, the low-entropy samples will be assigned with a large interval size.

Smoothing a sample will blur its details and often decrease its information. However, for a very simple image (with a low entropy), e.g., a handwritten digit, the smoothing may excessively eliminate its details, which are important to the classification task. Namely, the low-entropy image can't tolerate the blurring well from the perspective of classification. To this end, we use the entropy to decide whether the sample needs to be smoothed.

3.2 Evaluation Targets and Metrics

Target Models and Datasets. According to the three attack techniques presented in Section 2, we do our best to collect five available target models to explore the effectiveness of the proposed method. As shown in Table 1, the models cover all three attack techniques and refer to two datasets, MNIST [24] and ImageNet [25]. MNIST is a dataset of simple gray handwritten digits, while ImageNet is a large-scale dataset of labeled high-resolution images. We select the whole MINIST test dataset (10,000 digits) and seven classes of images (3,730 samples) of ImageNet for evaluation.

To prevent over-fitting, we determine hyper-parameters in a *cross validation* way. As listed in Table 2, these samples are divided into three parts, *training set*, *validation set* and *test set*. The training set is used to determine various detection parameters (e.g., the interval size, the entropy threshold, the choice of spatial smoothing filter, etc.), and the validation set is leveraged to verify whether the selected parameter setting is indeed effective. Besides, we only employ FGSM attack method in the parameter training phase, and the ε value is fixed to 0.2 for MNIST and 1/255 for ImageNet. After the parameters are determined, we will evaluate the performance of our method against all the three attack techniques on the test set.

TABLE 1: Target models.

Model	Dataset	Attack
M1 [36]	MNIST	FGSM
M2 [37]	MNIST	CW
CaffeNet [38]	ImageNet	DeepFool
GoogLeNet [38]	ImageNet	FGSM, DeepFool
Inception v3 [37]	ImageNet	CW

TABLE 2: Target datasets.

Dataset	Training	Validation	Test
MNIST	No. 0∼4499	No. 4500∼5499	No. 5500∼9999
ImageNet	Goldfish (648) Pineapple (520) Clock (455)	Jellyfish (618)	Zebra (503) Panda (501) Cab (485)

Evaluation Metrics. To explore some hyper-parameters of our method and evaluate the effectiveness, we adopt the *recall* rate, the *precision* rate and the *F1* score [27] to quantify the detection performance, which are defined as follows,

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$F1 = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{5}$$

where TP is the number of correctly detected adversarial examples (true positives), FN the number of adversarial examples that survive from our detection (false negatives), and FP the number of benign images that are detected as adversarial examples (false positives). The higher F1 score indicates the better overall detection performance. Ideally, if the method can detect all the adversarial examples without introducing any false positives, the F1 score will be 1.0.

Besides, to evaluate the capability of recovering the original classification of adversarial examples via our detection filter, we introduce a new metric *RTP* (short for *Recovered True Positives*). RTP indicates the number of such samples, which are successfully detected as adversarial and its filtered version can be correctly classified as its original class.

3.3 Computing Entropy

Our preliminary experiments indicate that the input image itself is also a parameter of determining effectual detection scheme. In other words, the scheme that works well on detecting low-resolution adversarial images may fail detecting high-resolution ones. In this work, we leverage the discrete entropy to distinguish images with different resolutions. Discrete entropy [39] is a powerful metric and has been proven useful in image processing [40], [41], [42]. Without loss of generality, for an M×N grayscale image with 256 pixel levels (0∼255), its entropy can be computed as equation (7). The frequency of pixel level i is denoted as f_i.

$$p_i = f_i \,/\, (M \times N) \qquad i = 0, 1, ..., 255 \tag{6}$$

TABLE 3: Uniform vs. non-uniform quantization.

Quantization	Time (s)	Recall	Precision	F1
Uniform	0.004	94.00%	100.00%	96.91%
Non-uniform	137.7	94.00%	51.65%	66.67%

$$\underset{\text{Cross mask}}{\frac{1}{9}\times\begin{bmatrix}0&0&1&0&0\\0&0&1&0&0\\1&1&1&1&1\\0&0&1&0&0\\0&0&1&0&0\end{bmatrix}}\quad\underset{\text{Diamond mask}}{\frac{1}{13}\times\begin{bmatrix}0&0&1&0&0\\0&1&1&1&0\\1&1&1&1&1\\0&1&1&1&0\\0&0&1&0&0\end{bmatrix}}\quad\underset{\text{Box mask}}{\frac{1}{25}\times\begin{bmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\end{bmatrix}}$$

Fig. 6: Different filter masks with size 5×5.

$$H = -\sum_{i=0}^{255} p_i log_2(p_i) \tag{7}$$

For a RGB color image, its entropy is the average of the entropies of its three color planes.

3.4 Scalar Quantization

Quantization is the process of representing a large (possibly infinite) set of values with a smaller (finite) one, e.g., mapping the real numbers to the integers. In image processing, quantization is often employed as a lossy compression technique by mapping a range of pixel intensities to a single representing one. In other words, reducing the number of colors of an image to cut its file size.

Scalar quantization is the most practical and straightforward approach to quantize an image. In scalar quantization, all inputs within a specified interval are mapped to a common value (called codeword), and the inputs in a different interval will be mapped to a different codeword. There are two types of scalar quantization, *uniform quantization* and *non-uniform quantization* [43]. In uniform quantization, the input will be separated into the same size intervals, while in non-uniform quantization, the intervals are usually of different sizes chosen with an optimization algorithm to minimize the distortion [44], [45]. A small-scale empirical study is performed on 100 MNIST digits with FGSM attack to decide which type of quantization can more effectively downgrade the effect of the perturbation. We adopt the algorithm in [44] to perform non-uniform quantization. The number of intervals is set to two for both uniform and non-uniform quantization. As Table 3 shows, uniform quantization can achieve much higher F1 score in detection and spend much less time. Therefore, in this study, we employ the uniform quantization technique to handle the sample.

To develop a uniform quantization, we also need to determine the number of intervals. We tried to use the samples of the training set to discover an appropriate interval number. Unfortunately, from the experiment results shown in Table 4, we can see that there is not a general interval setting for different samples. The best F1 score 96.43% for detecting MNIST adversarial examples is achieved under two-interval setting. However, for ImageNet examples, the best score 88.11% is got when using six-interval. To tackle this problem, we leverage the image entropy described in Section 3.3 to automatically apply an adaptive interval size for a given sample. We conduct a series of experiments on the training set to seek some entropy thresholds, which will be used to set the interval size. Finally, as shown in Table 5, we find a practicable solution. The number of intervals is set to two, four and six for the samples with different entropies (smaller than 4.0, between 4.0 and 5.0, and larger than 5.0) respectively. As Table 6 presents, the cross validation experiments also show that the adaptive scalar quantization strategy can achieve satisfying detection performance on both training and validation set.

However, the experiments also show that there is still room to improve the detection performance on high-entropy (larger than 5.0) samples. The F1 score is 85.80% for high-entropy samples but 95.68% for the others. In essence, scalar quantization is a kind of *point operation*. Subsequently, we further reduce perturbation for high-entropy samples by introducing the *neighborhood operation* technique. As Table 5 presents, we also use the entropy to determine whether the smoothing should be performed.

3.5 Spatial Smoothing Filter

The *spatial smoothing filter* is one of the most classic techniques for noise reduction. The idea behind it is to modify the value of the pixels in an image based on a local neighborhood of the pixels. As a result, the sharp transitions in pixel intensities, often brought by noise, are reduced in the target image. In linear smoothing filtering, the filtered image $f'(m, n)$ is the *convolution* of the original image $f(m, n)$ with a filter mask $w(m, n)$ as follows.

$$f'(m,n) = (w*f)(m,n) = \sum_{s=-a}^{a}\sum_{t=-b}^{b} w\,(s,t)\,f\,(m-s,n-t) \tag{8}$$

The filter mask determines the smoothing effect. Fig. 6 presents three common 5×5 averaging filter masks. With a mask, the intensity of a pixel is replaced with the standard average of the intensities of the pixels which are weighted by 1 in its neighborhood. After filtered, the target image is blurred and small details are removed from it. However, from the viewpoint of image classification, the objects of interest may be highlighted and easy to detect.

To determine a practicable filter, we conduct a number of experiments with different shape and size of masks on the 2,015 high-entropy training samples. From the results listed in Table 7, it can be concluded that there are five acceptable candidates, i.e., 5×5 cross, 7×7 cross, 5×5 diamond, 7×7 diamond, and 5×5 box masks. The cross validation experiments (see Table 8) also indicate that these five masks are effective on our validation set. In the next subsection, we will present the complete detection filter based on scalar quantization and spatial smoothing.

3.6 Detection Filter

In practice, the smoothing may bring an excessive blurring to some samples and produce new false positives. To this end, we design a combination filter based on the two above techniques rather than simply concatenating them together.

As discussed in Section 2, the attacker often wants the perturbation introduced in the adversarial example as small as possible to make it imperceptible. In other words, the

TABLE 4: Performance of detecting FGSM adversarial examples with different scalar quantization schemes.

Dataset	Metric	Number of Intervals								
		2	3	4	5	6	7	8	9	10
MNIST	Recall	**93.86%**	86.48%	93.04%	13.26%	7.25%	24.95%	42.03%	50.62%	52.35%
	Precision	**99.14%**	99.66%	99.90%	99.51%	99.11%	99.74%	99.77%	100.00%	100.00%
	F1 Score	**96.43%**	92.60%	96.35%	23.40%	13.51%	39.92%	59.15%	67.22%	68.72%
ImageNet	Recall	94.35%	93.07%	91.26%	86.59%	**85.15%**	80.71%	76.56%	68.50%	66.84%
	Precision	57.51%	72.14%	82.05%	88.32%	**91.28%**	93.54%	94.60%	94.79%	95.27%
	F1 Score	71.46%	81.28%	86.41%	87.44%	**88.11%**	86.65%	84.63%	79.53%	78.57%

TABLE 5: The denoising strategies for different entropies.

Entropy	Quantization Intervals	Smoothing?
<4	2	NO
4 ∼5	4	NO
>5	6	YES

TABLE 6: Detection performance of the adaptive quantization strategy.

Dataset	TP	FN	FP	Recall	Precision	F1 Score
Training	3482	370	146	90.39%	95.98%	93.10%
Validation	939	81	48	92.06%	95.14%	93.57%

perturbation to the pixel intensity is often limited in a small range. If the intensity of a pixel is blurred too much by the smoothing filter, the smoothing might be unnecessary. Based on the above intuition, our detection filter for high-entropy samples is defined by the following equation, $f'(m,n)$ =

$$\begin{cases} f_{SQ}(m,n), \ \textbf{if} \mid f_{SQ}(m,n) - f(m,n) \mid \\ \qquad\qquad \leq \ \mid f_{SQ-SF}(m,n) - f(m,n) \mid \\ f_{SQ-SF}(m,n), \ \textbf{else} \end{cases} \tag{9}$$

where $f_{SQ}(m,\ n)$ is the quantized original image and $f_{SQ-SF}(m,\ n)$ is the smoothed quantized image. For a given pixel $(a,\ b)$ of the input sample $f(m,\ n)$, the output pixel value $f'(a,\ b)$ will be replaced with its quantization $f_{SQ}(a,\ b)$ when the distance between the quantization $f_{SQ}(a,\ b)$ and the original pixel value $f(a,\ b)$ is smaller than the one between $f_{SQ-SF}(a,\ b)$ and $f(a,\ b)$; otherwise, it will be set to $f_{SQ-SF}(a,\ b)$.

The complete framework of our detection method is ready now. In the first place, the image entropy is leveraged to automatically adjust detection parameters for each sample. For low-entropy (smaller than 4.0) and mid-entropy (between 4.0 and 5.0) samples, we only apply two-interval and four-interval uniform quantization to them respectively; for high-entropy (larger than 5.0) samples, we first quantize them with six-interval quantization, and then smooth their quantized version as illustrated in Equation 9. To determine the final smoothing filter from the five candidates, we conduct a series of experiments and find that 7×7 cross mask is the best one when integrating it with scalar quantization. As Table 9 shows, the final detection filter can achieve satisfying detection performance on both training and validation set.

Note that the proposed method is transparent to the target model. In practice, the detection filter can be directly integrated with any off-the-shelf model as a sample preprocessor. The target model can be kept unchanged.

4 Evaluation

In this section, we evaluate the proposed method on our test set, consisting of 4,500 MNIST digits and 1,489 ImageNet images of three classes (*Zebra*, *Panda* and *Cab*). Additionally, all the three attack techniques, i.e., FGSM, DeepFool and CW, are employed to inspect the effectiveness of our method. It should be pointed out that all the test samples and the latter two attack techniques (DeepFool and CW) are not involved in the design phase of our method. As mentioned in Section 1, the proposed method will be evaluated in in both defense-unaware and defense-aware scenarios.

4.1 Defense-unaware Attack Detection

In the defense-unaware scenario, the adversary is ignorant of the proposed method, and can only generate adversarial examples on the original model.

4.1.1 Detecting FGSM Examples

For FGSM attack, the ε value is an important and adjustable parameter. In general, the larger the ε is, the more adversarial examples FGSM can successfully crafted. However, an excessive ε is likely to introduce noticeable perturbation and be easily spotted by a human. To evaluate the capability of our method for detecting adversarial examples with different magnitudes of perturbations, the adversarial examples are crafted with some acceptable ε values. It is set between 0.1 to 0.4 for MNIST digits and 1/255 or 2/255 for ImageNet images. As listed in Table 10 (rows 1∼6), the proposed method can achieve an average F1 score of 95.37% in detecting all FGSM examples. Surprisingly, the classifications of 94.16% adversarial images can be correctly recovered by the detection filter. Note that, the detection performance is obtained without using any prior knowledge about adversarial examples.

4.1.2 Detecting DeepFool Examples

We use the algorithm provided in [46] to generate DeepFool examples against GoogLeNet and CaffeNet [38]. The detection performance is presented in Table 10 (rows 7 and 8). Although DeepFool can produce a smaller perturbation than FGSM, the proposed method is still effective. Specifically, the proposed method can achieve high F1 scores of 93.85% and 95.95% with respect to GoogLeNet and CaffeNet, respectively. Besides, the classifications of 95.02% DeepFool examples can be successfully recovered by our filter.

TABLE 7: Performance of detecting high-entropy FGSM adversarial examples with different spatial smoothing filters.

Filter Mask	Cross Mask				Diamond Mask				Box Mask			
	3×3	$\mathbf{5\times5}$	$\mathbf{7\times7}$	9×9	3×3	$\mathbf{5\times5}$	$\mathbf{7\times7}$	9×9	3×3	$\mathbf{5\times5}$	7×7	9×9
Recall	56.43%	**84.05%**	**90.03%**	91.03%	56.34%	**85.54%**	**90.93%**	92.22%	74.18%	**91.53%**	92.12%	93.22%
Precision	90.85%	**91.83%**	**86.00%**	80.23%	90.85%	**88.27%**	**81.57%**	75.39%	88.15%	**82.41%**	74.28%	66.08%
F1 Score	69.62%	**87.77%**	**87.97%**	85.29%	69.62%	**86.89%**	**86.00%**	82.96%	80.56%	**86.73%**	82.24%	77.34%

TABLE 8: Performance of the five superior spatial smoothing filters with the validation set.

Filter Mask	Cross Mask		Diamond Mask		Box Mask
	5×5	7×7	5×5	7×7	5×5
Recall	80.33%	84.78%	82.20%	87.12%	85.95%
Precision	90.74%	83.60%	86.67%	78.98%	80.48%
F1 Score	85.22%	84.19%	84.38%	82.85%	83.13%

TABLE 9: Performance of the proposed detection filter on detecting FGSM adversarial examples.

Dataset	TP	FN	FP	Recall	Precision	F1
Training	3324	266	108	92.59%	96.85%	94.67%
Validation	1028	61	35	94.40%	96.71%	95.54%

4.1.3 Detecting CW Examples

Among the three techniques, CW attack is the strongest one. It is an optimization-based algorithm which can seek out as small as possible perturbations. Carlini and Wagner [26] demonstrated that CW attack can find closer adversarial examples than the other attack techniques. For example, CW L_2 attack can find adversarial examples with at least two times lower distortion than FGSM. Besides, they proved that CW attack can craft adversarial examples with very high success rate. The #*F* column of Table 10 further demonstrates the statement.

In the experiments, we employ CW L_2 and L_∞ attack to generate well-disguised adversarial examples to evaluate our method. However, CW attack is much more expensive than other attack techniques. Generating an ImageNet adversarial example with L_2 and L_∞ can take about 5 and 40 minutes respectively. For this reason, we only picked the last 1,000 MNIST digits and 100 randomly selected ImageNet images from the test set as experiment dataset.

CW L_2 attack can generate adversarial examples under various confidences. To comprehensively evaluate our method, when generating L_2 examples, we use different κ values [26] to control the confidence of generated examples. When $\kappa = 0.0$, the adversarial examples will be with low-confidence. As κ increases, the confidence will increase accordingly. In the experiments, we set κ to 0.0, 0.5, 1.0, 2.0 and 4.0, and generate L_2 examples with confidence from about 50% to over 90%. As listed in Table 10 (rows 9∼20), our method illustrates excellent detection performance. It can achieve an average F1 score of 98.80%. In addition, the classifications of 93.94% CW examples can be recovered successfully. By the way, detecting a sample with the proposed method only takes less than one second. The introduced overhead is totally negligible compared with the time consumption of generating a CW example.

4.1.4 The Proposed Method vs. Other Detections

Recently, Carlini and Wagner [28] presented a very comprehensive survey on the effectiveness of detection methods. They collect ten recent detection methods they could find and re-implemented seven of them, which didn't release any source code. The CW L_2 attack algorithm was employed to craft adversarial examples to test the methods under different attack scenarios. Their study involved a great amount of work and provided a solid basis for estimate our detection method. In this study, we follow the methodology of Carlini and Wagner [28] and leverage their result to compare our method with related ones in the defense-unaware and defense-aware scenarios.

One [18] of the ten methods is other type of detection and beyond the scope of this paper. All the rest nine ones are used in the comparison. As listed in the second column of Table 11, five target detection methods [17], [19], [48], [49], [50] can effectively detect adversarial examples under defense-unaware scenario. Carlini and Wagner [28] provided exact recall rates for three of them [19], [49], [50]. However, the other two target methods [17], [47] are less effective and cannot achieve a high recall rate. The remaining two [19], [47] have been proven to be ineffective when facing CW L_2 attack.

We adopt the same configurations as [28], i.e., using the default κ value setting (0.0), to generate L_2 examples. The detection performance of our method is presented in Table 10 (row 9). Our method can achieve a high recall rate of 98.89% and a low false positive rate of 0.90%. Compared with the existing methods, we can conclude that our method outperforms them in the defense-unaware scenario.

4.2 Defense-aware Attack detection

In the defense-aware scenario, the adversary knows the technique details of the detection method. When launching an attack, he or she can leverage the knowledge to fool the classifier as well as the detector. As done in [28], we launch a defense-aware CW L_2 attack on 1,000 MNIST test samples.

Specifically, to evade our detection, the adversary needs to generate an adversarial example x, which satisfies the constraint $C(x) == C(T(x))$. We modify the original CW L_2 algorithm to get such examples. The constraint is introduced in the iteration of seeking an effectual adversarial example. This ensures the output image of every optimization and its denoised version are classified as the same thing. If the algorithm can output an adversarial example successfully, it will evade our detection method. The experiments show that there are 32.63% of the samples cannot be perturbed successfully to the target model equipped with our detection. In other words, the attack success rate is 67.37%.

Carlini and Wagner [28] also presented defense-aware attack experiments for the seven detection methods except

TABLE 10: Detection results of the proposed method. #F denotes the number of images that cannot be perturbed to adversarial ones.

No.	Attack/Model	Dataset	#F	TP	FN	FP	RTP	RTP%	Recall	Precision	F1
1	FGSM (ε=0.1)/M1	MNIST	4026	410	40	24	398	97.07%	91.11%	94.47%	92.76%
2	FGSM (ε=0.2)/M1	MNIST	1910	2467	106	32	2430	98.50%	95.88%	98.72%	97.28%
3	FGSM (ε=0.3)/M1	MNIST	455	3856	172	32	3768	97.71%	95.73%	99.18%	97.42%
4	FGSM (ε=0.4)/M1	MNIST	132	4078	273	32	3820	93.67%	93.73%	99.22%	96.40%
5	FGSM (ε=1/255)/GoogLeNet	ImageNet	270	841	98	88	718	85.37%	89.56%	90.53%	90.04%
6	FGSM (ε=2/255)/GoogLeNet	ImageNet	119	860	230	89	647	75.23%	78.90%	90.62%	84.36%
7	DeepFool/GoogLeNet	ImageNet	405	725	42	53	690	95.17%	94.52%	93.19%	93.85%
8	DeepFool/CaffeNet	ImageNet	96	900	29	47	854	94.89%	96.88%	95.04%	95.95%
9	**CW L_2 (κ=0.0)/M2**	**MNIST**	**0**	**984**	**11**	**9**	**919**	**93.39%**	**98.89%**	**99.09%**	**98.99%**
10	CW L_2 (κ=0.5)/M2	MNIST	0	984	11	9	920	93.50%	98.89%	99.09%	98.99%
11	CW L_2 (κ=1.0)/M2	MNIST	0	983	12	9	913	92.88%	98.79%	99.09%	98.94%
12	CW L_2 (κ=2.0)/M2	MNIST	0	979	16	9	897	91.62%	98.39%	99.09%	98.74%
13	CW L_2 (κ=4.0)/M2	MNIST	0	959	36	9	866	90.30%	96.38%	99.07%	97.71%
14	CW L_2 (κ=0.0)/Inception v3	ImageNet	2	98	0	8	91	92.86%	100%	92.45%	96.08%
15	CW L_2 (κ=0.5)/Inception v3	ImageNet	0	100	0	2	98	98.00%	100%	98.04%	99.01.%
16	CW L_2 (κ=1.0)/Inception v3	ImageNet	0	100	0	2	98	98.00%	100%	98.04%	99.01.%
17	CW L_2 (κ=2.0)/Inception v3	ImageNet	0	100	0	2	98	98.00%	100%	98.04%	99.01.%
18	CW L_2 (κ=4.0)/Inception v3	ImageNet	0	100	0	2	98	98.00%	100%	98.04%	99.01.%
19	CW L_∞/M2	MNIST	0	991	7	1	991	100%	99.30%	99.90%	99.60%
20	CW L_∞/Inception v3	ImageNet	25	75	0	2	73	97.33%	100.00%	97.40%	98.68%
Total/Average			7440	20590	1083	461	19387	94.16%	95.00%	97.81%	96.39%

TABLE 11: The proposed method vs. other detections.

Method	Defense-unaware	Defense-aware
	Detection Performance	Attack Success?
MMD [19]	"fails to detect" [28]	-
Cascade Classifier [47]	92.00% false positive	-
Network Uncertainty [17]	75.00% recall	98.00%
3×3 Filter [47]	80.00% recall	YES
KDE [17]	"able to detect" [28]	YES
PCA [48]	"does detect" [28]	YES
Dimensionality Reduction [49]	97.00% recall	YES
Adversarial Training [50]	98.00% recall	100%
Adversarial Retraining [19]	98.50% recall	100%
The proposed method	**98.89% recall 0.90% false positive**	**67.37%**

the two that already fail in defense-unaware attacks. As done in Section 4.1.4, we also use their results to make a comparison with other methods. From the third column of Table 11, we can see that three methods [17], [19], [50] become completely ineffective. An adversary can successfully generate adversarial examples for almost all of samples. And all the other four methods [17], [47], [49], [50] are also "broken" [28]. By comparison, our method can remarkably downgrade the success rate from nearly 100% to 67.37%.

As discussed in [26] and [28], it is still an open problem to construct defenses to defense-aware attacks. Based on the above experiments, we believe that our method effectively raises the bar for adversaries to launch successful attacks.

4.3 Summary

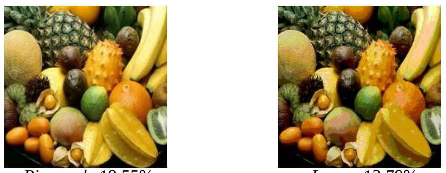
All in all, 43,346 samples are used to evaluate our method in the defense-unaware experiments, half of them are adversarial and half are benign. In detecting those adversarial examples generated by the three attack techniques, we achieve an overall recall of 95.00% and an overall precision of 97.81%, resulting in a high overall F1 score of 96.39%. Note that our detection method inevitably leads to some false positives. For the 21,673 benign samples, our method causes 461 samples (2.13%) to be misclassified, which is an acceptable performance degradation in security critical scenarios. In addition, the classifications of 94.16% adversarial examples can be recovered by the proposed filter.

In practice, it is very difficult to effectively detect defense-aware attacks, as demonstrated in previous studies. All existing detection methods were broken by defense-aware CW attacks. As Section 4.2 shows, our method certainly raises the bar for adversaries, and is a promising mitigation to defense-aware attacks.

Furthermore, our method can be deployed directly to the target model without modifying or retraining it, and even without any prior knowledge of attack techniques. In fact, we only use FGSM algorithm for determining detection filters, but the selected filter can effectively defeat the other two much stronger attacks, i.e., DeepFool and CW.

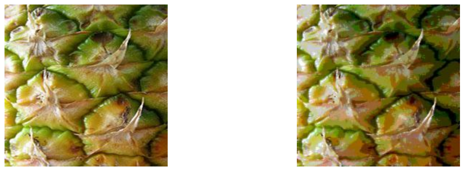
5 Discussion and Limitations

False Positives and False Negatives. In principle, the performance of our detection method is closely related to the classification capacity of target classifiers. Some false positives and false negatives are caused by the ambiguous images, which are essentially hard to classify for the target classifier. As shown in Fig. 7, an image consisting of various fruits is labeled as *Pineapple*, but GoogLeNet can tell that


Pineapple 19.55%
Lemon 10.85%
Jackfruit 9.43%
Benign Sample

Lemon 13.79%
Pineapple 13.03%
Jackfruit 8.03%
Filtered Sample

Fig. 7: A false positive caused by the ambiguous sample


Sea anemone 19.76%
Lemon 12.32%
Fig 8.17%
Adversarial Sample

Sea anemone 21.83%
Coral reef 14.65%
Jellyfish 9.29%
Filtered Adversarial Sample

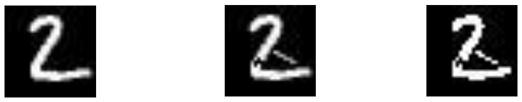
Fig. 8: A false negative caused by the ambiguous sample

with only 19.55% confidence. This is really not a strong prediction. The sample is also considered as a *Lemon* with 10.85% confidence and a *Jackfruit* 9.43%. After being denoised by our filter, the image is misclassified as *Lemon* and results in a false positive. However, we think that neither the model nor the proposed detection filter is to blame for the false positive, but the ambiguity within the image is. The confusing images cannot only result in false positives, but also false negatives. Take an adversarial example generated with FGSM as example, the image shown in Fig. 8 is perturbed from *Pineapple* to *Sea Anemone* but only with 19.76% confidence. GoogLeNet gives a weak prediction for it. And our detection method also fails to detect this adversarial example and produces a false negative.

There are quite a few ambiguous images like the two examples in our experiment set, which brings down the precision rate as well as the recall rate. For the ambiguous samples, inspecting more predict classes (e.g. top five predictions) might be necessary rather than just the top one with the highest confidence.

Perceptible Perturbations. Some attack techniques, such as CW L_0 [26] and *Jacobian-based saliency map approach* (JSMA) [15], may introduce the large-amplitude perturbation. Such attack techniques limit the number of altered pixels, but not the amplitude of pixels. Consequently, as illustrated in Fig. 9, the obtained adversarial example may present easy-to-notice distortions. It can be easily spotted by a human. However, it can still be exploited to launch an effective attack when the human interaction is not in consideration. Some other adversarial images with perceptible perturbations can be found in [15]. In principle, it is very difficult to properly reduce the effect of the heavy perturbation only with the filtering technique without compromising the semantics of the original image. To develop an effective technique to detect this kind of example is beyond the scope of this paper but will be our future research.

Other Image Processing Techniques. There are a number of other image processing techniques in addition to the ones adopted in our method. Some of them may can be leveraged to further improve our detection method, such as *Rényi entropy* [51], *image segmentation* [43], etc. For example, we can segment an adversarial example into some regions according to the connectivity among pixels to find such a region that possesses as much as possible information. From it, we have a good chance to restore the correct classification when the perturbation is isolated in other regions. In this way, the adversarial example shown in Fig. 9 may can be detected. In the future, we plan to investigate other im-


Two
Original Sample

Three
Adversarial Sample

Three
Filtered Adversarial Sample

Fig. 9: A perceptible adversarial example generated with CW L_0 attack.

age processing techniques to develop a more sophisticated detection method, especially for detecting the adversarial example with large-amplitude perturbations.

6 Related Work

Many existing studies have paid much attention to the security of classifiers, and the arm race between adversaries and defenders will never end.

Attacks on Traditional Classifiers. Many studies have investigated the security of traditional machine learning methods [52] and proposed some attack methods. Lowd and Meek conduct an attack that minimizes a cost function [53]. They further propose attacks against statistical spam filters that add the words indicative of non-spam emails to spam emails [54]. The same strategy is employed in [55]. In [56], a methodology, called *reverse mimicry*, is designed to evade structural PDF malware detection systems. In [57], an online learning-based system, PDF_{RATE}, was used as a case to investigate the effectiveness of evasion attacks. The study reconstructs a similar classifier through training one of the publicly available datasets by a few deduced features, and then evades PDF_{RATE} by insertion of dummy content into PDF files. In [58], an algorithm is proposed for evasion of classifiers with differentiable discriminant functions. Liang et al. [59] demonstrated that client-side classifiers are also vulnerable to evasion attacks.

Xu et al. [60] presented a general approach to find evasive variants by stochastically manipulate a malicious sample seed. The experiments showed that the effectual variants can be automatically generated to against two PDF malware classifiers, i.e., PDF_{RATE} and Hidost.

Fredrikson et al. [61], [62] developed a new form of model inversion attack which can infer sensitive features used in decision tree models and recover images from some facial recognition models by exploiting confidence values revealed by the target models. The proposed attack may cause serious privacy disclosure problems [61]. More model inversion attacks can be found in [62].

Defenses for Traditional Classifiers. Many countermeasures against evasion attacks have been proposed, such as using game theory [63], [64] or probabilistic models [65], [66] to predict attack strategy to construct more robust classifiers, employing multiple classifier systems (MCSs) [67], [68], [69] to increase the difficulty of evasion, and optimizing feature selection [70], [71] to make the features evenly distributed.

Game-theoretical approaches [63], [64] model the interactions between the adversary and the classifier as a game. The adversary's goal is to evade detection by minimally manipulating the attack instances, while the classifier is retrained to correctly classify them.

MCSs [67], [68], [69], as the name suggests, uses multiple classifiers rather than only one to improve classifier's robustness. The adversary who wants to effectively evade the classification has to fight with more than one classifier.

Kantchelian et al. [72] present family-based ensembles of classifiers. In particular, they trained an ensemble of classifiers, one for each family of malware. By combining classifications, it will be determined whether an unknown binary is malware, and if it is, which family it belongs to. What's more, they also demonstrate the importance of human operators in adversarial environments.

In [70], the method *weight evenness* via feature selection optimization is proposed. By appropriate feature selection, the weight of every feature is evenly distributed, thus the adversary has to manipulate a larger number of features to evade detection. In [71], the features are reweighted inversely proportional to their corresponding importance, making it difficult for the adversary to exploit the features.

Unfortunately, these attack and defense techniques for traditional classifiers cannot be directly applied to DNNs. Along with the prevalence of DNNs, researchers have begun to pay close attention to the security of DNNs.

Attacks on DNNs. Recently, researchers have begun to attack DNN-based classifiers through crafting adversarial examples. There are various methods to generate adversarial examples against DNNs in various fields, not limited in computer vision [10], [11], [12], but also speech recognition [14], text classification [73] and malware detection [74].

Kereliuk et al. [14] proposed a method to craft adversarial audio examples using the gradient information of the model's loss function. Through the application of minor perturbations to the input magnitude spectra, they can effectively craft an adversarial example. Text as discrete data is sensitive to perturbation. Liang et al. [73] proposed a method to craft adversarial text examples. Three perturbation strategies, namely *insertion*, *modification*, and *removal*, are designed to generate an adversarial example for a given text. By computing the cost gradients, what should be inserted, modified or removed, where to insert and how to modify are determined effectively. By elaborately dressing a text sample, the adversary can modify the classification to any other classes while still keeps the meaning unchanged. Grosse et al. [74] presented a method to craft adversarial examples on neural networks for malware classification, by adapting the method originally proposed in [15].

In this paper, we focus on the detection of adversarial images. We believe that our method can be applied to detect adversarial examples for audio, which is also a kind of continuous data. But the proposed technique cannot be applied to discrete data, such as the adversarial text and malware. The new method need to be developed for effectively detecting them.

Note that there are two recent studies focus on crafting adversarial examples in the physical world. Kurakin et al. [75] demonstrated that the adversarial images obtained from a cell-phone camera can still fool an ImageNet classifier. Sharif et al. [76] presented an attack method to fool facial biometric systems. They showed that with some well-crafted eyeglass frames, a subject can dodge recognition or impersonate others.

Besides, Shokri et al. [77] developed a novel black-box membership inference attack against machine learning models, including DNN and non-DNN models. Given a data record, the attacker can determine whether it is in the target model's training dataset. For health-care datasets, such information leakage is unacceptable.

Improve the Robustness of Deep Networks. The adversarial training [10], [11], [14], [15], [78] is a straightforward defense technique to improve the robustness of target models. Retraining models by adding as many as possible adversarial examples can bring more challenges for attackers to find new adversarial examples.

Wang et al. [79] integrated a data transformation module right in front of a standard DNN to improve the model's resistance to adversarial examples. This data transformation module leverages non-parametric dimensionality reduction methods, and projects all the input samples into a new representation before passing the inputs to the target DNN in training and testing. Wang et al. [80] also proposed another method, named *random feature nullification*, for constructing adversary resistant DNNs. In particular, it randomly nullifies or masks features within input samples in both the training and testing phase. Such nullification makes a DNN model non-deterministic and then improves model's resistance to adversarial examples.

The proposed method is compatible with the above defense techniques. Defenders can still use our method in the enhanced model to get a better performance.

Detection Techniques. Some studies also focus on detecting adversarial examples directly.

Xu et al. in a very recent study [20] proposed a method, called *Feature Squeezing*, to detect adversarial examples in a similar way as ours. They explore two approaches to squeeze the features of an image: reducing the color bit depth of each pixel and smoothing it using a spatial filter. Their system identifies the adversarial examples by measuring the disagreement among the prediction vectors of the original and squeezed examples. Their experiments illustrated high performance was achieved when detecting FGSM adversarial examples in a MNIST model. However, a predefined threshold is required for determining how much disagreement indicates the current sample is adversarial. In their experiments, half of the examples are used to train the threshold that can produce the best detection accuracy on training examples. This means the defender must have a sufficient number of adversarial examples generated with potential attack techniques. As a result, the method works well only when the attack technique is known but less effective when facing unknown attacks. Moreover, in principle, for different datasets, models or attacks, the thresholds need to be retrained to achieve acceptable performance. By introducing the image entropy, we implement an adaptive detection method and can be directly applied to different models, datasets and attack techniques with the same setting and without requiring any prior knowledge of attacks.

Grosse et al. [19] put forward a defense to detect adversarial examples using statistical tests. The method requires a sufficient large group of adversarial examples and benign examples to estimate their data distribution. However, the statistical test method cannot be directly applied to detect individual examples, making it less useful in practice. For this reason, Grosse et al. further propose a new method by adding an additional class (e.g., adversarial class) to the model's output and retraining the model to classify adversarial examples as the new class. Gong *et al.* [50] proposed a similar adversarial retraining-based detection method. In the method, they trained a binary model to determine whether a given input is adversarial. Bhagoji et al. [49] leveraged the principle component analysis (PCA) technique to reduce the dimensionality of the images. Then instead of training on the original images, they trained a classifier on images which had been processed with dimensionality reduction. Hendrycks and Gimpel [48] used a large number of adversarial examples to train a detector to identify unknown adversarial examples. A small ″detector″ subnetwork is trained on the binary classification task of distinguishing benign samples from adversarial perturbations.

To a large extent, the performance of the above four detection techniques [19], [48], [49], [50] also depends on how much effectual adversarial examples are available.

Li *et al.* [47] built a cascade classifier and integrated the classifier with the original model. By checking each output of the original model's inner convolutional layers, they can determine whether a input sample is adversarial. Feinman et al. [17] devised two novel features to detect adversarial examples based on the idea that adversarial examples deviate the true data manifold. They introduced density estimates to measure the distance between an unknown input sample and a set of benign samples. The method is computationally expensive and may be less effective in detecting adversarial examples which are very close to benign samples.

In summary, compared with the above defense techniques, our method possesses three advantages. First, our method is adaptive. The detection strategy is automatically adjusted for individual samples. The input image will be suitably filtered according to its entropy before being fed to the classifier. This improves the detection performance greatly. Second, the method is not attack-specific and thus holds great promise for detecting unknown attacks. As the experiments demonstrated, the detection parameters tuned with respect to a weaker attack (FGSM) is also applicable to stronger attacks (DeepFool and CW). Finally, our method is easier to deploy. The method can be directly integrated with a trained model, without retraining or modifying the model.

7 Conclusion

Many efforts have been paid to use various techniques to defend or detect the adversarial image examples in DNNs. However, the prior knowledge of attack techniques or the modification to the target model is often required. This paper presents a straightforward and effective adversarial image examples detection method. The adversarial perturbations are regarded as a kind of noise and the proposed method is implemented as a filter to reduce their effect. The image entropy is used to automatically adjust the detection strategy for specific samples. Our method provides two important features (1) without requiring the prior knowledge about attacks and (2) can be directly integrated into unmodified models.

We evaluate our method in both defense-unaware and defense-aware scenario. The experiments show that our method can achieve a high F1 score in detecting the adversarial examples generated by the different attack techniques and targeting different models in the defense-unaware scenario, and can effectively raise the bar for adversaries in the defense-aware scenario. Compared with existing detection methods, our method can perform better in both the two attack scenarios. Note that our method is also compatible with other defense techniques. A better performance can be achieved by combining them together.

Our research demonstrates that the adversarial images can be effectively analyzed with classic image processing techniques. In the future, we will investigate more image processing techniques to find more effective and practicable detection techniques, especially for defense-aware attacks.

Acknowledgments

The authors would like to thank the anonymous reviewers for their insightful comments on the preliminary version of this paper. The work is supported by National Natural Science Foundation of China (NSFC) under grants 91418206, 61802413, 61672523, and 61472429, National Science and Technology Major Project of China under grant 2012ZX01039-004.

References

[1] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[2] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proceedings of the 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2010, pp. 253–256.

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[4] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[5] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[6] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 160–167.

[7] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proceedings of Advances in Neural Information Processing Systems*, 2015, pp. 649–657.

[8] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1891–1898.

[9] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proceedings of Advances in Neural Information Processing Systems*, 2014, pp. 1988–1996.

[10] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proceedings of the 2015 International Conference on Learning Representations*, 2015.

[11] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2574–2582.

[12] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proceedings of the 2014 International Conference on Learning Representations*, 2014.

[13] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. ACM, 2017, pp. 506–519.

[14] C. Kereliuk, B. L. Sturm, and J. Larsen, "Deep learning and music adversaries," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 2059–2071, 2015.

[15] 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 (EuroS&P)*. IEEE, 2016, pp. 372–387.

[16] 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 (S&P)*. IEEE, 2016, pp. 582–597.

[17] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting adversarial samples from artifacts," *arXiv preprint arXiv:1703.00410*, 2017.

[18] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On detecting adversarial perturbations," *arXiv preprint arXiv:1702.04267*, 2017.

[19] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. McDaniel, "On the (statistical) detection of adversarial examples," *arXiv preprint arXiv:1702.06280*, 2017.

[20] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," in *Proceedings of Network and Distributed System Security Symposium*, 2018.

[21] I. Goodfellow, H. Lee, Q. V. Le, A. Saxe, and A. Y. Ng, "Measuring invariances in deep networks," in *Proceedings of Advances in Neural Information Processing Systems*, 2009, pp. 646–654.

[22] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[23] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[24] "The MNIST database of handwritten digits," http://yann.lecun.com/exdb/mnist.

[25] D. Jia, D. Wei, S. Richard, L.-J. Li, L. Kai, and F.-F. Li, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 248–255.

[26] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proceedings of the 2017 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2017, pp. 39–57.

[27] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," 2011.

[28] 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*. ACM, 2017, pp. 3–14.

[29] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. rndi, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Th European Conference on Machine Learning and Knowledge Discovery in Databases*, 2017, pp. 387–402.

[30] Y. Liu, S. Ma, Y. Aafer, W. C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," in *Network and Distributed System Security Symposium*, 2017.

[31] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, p. 115, 2017.

[32] A. Singla, L. Yuan, and T. Ebrahimi, "Food/non-food image classification and food categorization using pre-trained googlenet model," in *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management*. ACM, 2016, pp. 3–11.

[33] Q. V. Le, "Building high-level features using large scale unsupervised learning," in *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 8595–8598.

[34] J.-S. Lee, "Digital image enhancement and noise filtering by use of local statistics," *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 165–168, 1980.

[35] D.-Y. Tsai, Y. Lee, and E. Matsuyama, "Information entropy measure for evaluation of image quality," *Journal of digital imaging*, vol. 21, no. 3, pp. 338–347, 2008.

[36] N. Papernot, I. Goodfellow, R. Sheatsley, R. Feinman, and P. McDaniel, "cleverhans v1.0.0: an adversarial machine learning library," *arXiv preprint arXiv:1610.00768*, 2016.

[37] "Robust evasion attacks against neural network to find adversarial examples," https://github.com/carlini/nn_robust_attacks.

[38] "BVLC models," https://github.com/BVLC/caffe/tree/master/models.

[39] R. M. Gray, *Entropy and information theory*. Springer Science & Business Media, 2011.

[40] C. Wang and Z. Ye, "Brightness preserving histogram equalization with maximum entropy: a variational perspective," *IEEE Transactions on Consumer Electronics*, vol. 51, no. 4, pp. 1326–1334, 2005.

[41] J.-H. Yoo, S.-Y. Ohm, and M.-G. Chung, "Maximum-entropy image enhancement using brightness mean and variance," *Journal of Internet Computing and Services*, vol. 13, no. 3, pp. 61–73, 2012.

[42] B. S. Min, D. K. Lim, S. J. Kim, and J. H. Lee, "A novel method of determining parameters of clahe based on image entropy," *International Journal of Software Engineering and Its Applications*, vol. 7, no. 5, pp. 113–120, 2013.

[43] R. C. Gonzalez and R. E. Woods, "Digital image processing," 2012.

[44] M. Srivastava and P. K. Panigrahi, "Non-uniform quantization of detail components in wavelet transformed image for lossy jpeg2000 compression," *arXiv preprint arXiv:1210.8165*, 2012.

[45] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy, "A study of the effect of jpg compression on adversarial images," *arXiv preprint arXiv:1608.00853*, 2016.

[46] "A simple and accurate method to fool deep neural networks," https://github.com/lts4/deepfool.

[47] X. Li and F. Li, "Adversarial examples detection in deep networks with convolutional filter statistics," *CoRR, abs/1612.07767*, vol. 7, 2016.

[48] D. Hendrycks and K. Gimpel, "Early methods for detecting adversarial images," *arXiv preprint arXiv:1608.00530*, 2016.

[49] A. N. Bhagoji, D. Cullina, and P. Mittal, "Dimensionality reduction as a defense against evasion attacks on machine learning classifiers," *arXiv preprint arXiv:1704.02654*, 2017.

[50] Z. Gong, W. Wang, and W.-S. Ku, "Adversarial and clean data are not twins," *arXiv preprint arXiv:1704.04960*, 2017.

[51] B. Chen and J.-j. Zhang, "On short interval expansion of rényi entropy," *Journal of High Energy Physics*, vol. 11, p. 164, 2013.

[52] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?" in *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*. ACM, 2006, pp. 16–25.

[53] D. Lowd and C. Meek, "Adversarial learning," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 2005, pp. 641–647.

[54] ——, "Good word attacks on statistical spam filters." in *Proceedings of the 2nd Conference on Email and Anti-Spam*, 2005.

[55] B. Nelson, M. Barreno, F. J. Chi, A. D. Joseph, B. I. Rubinstein, U. Saini, C. A. Sutton, J. D. Tygar, and K. Xia, "Exploiting machine learning to subvert your spam filter." *LEET*, vol. 8, pp. 1–9, 2008.

[56] D. Maiorca, I. Corona, and G. Giacinto, "Looking at the bag is not enough to find the bomb: an evasion of structural methods for malicious pdf files detection," in *Proceedings of the 8th ACM*

SIGSAC symposium on Information, computer and communications security. ACM, 2013, pp. 119–130.

[57] P. Laskov *et al.*, "Practical evasion of a learning-based classifier: A case study," in *Proceedings of the 2014 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2014, pp. 197–211.

[58] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2013, pp. 387–402.

[59] B. Liang, M. Su, W. You, W. Shi, and G. Yang, "Cracking classifiers for evasion: A case study on the google's phishing pages filter," in *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016, pp. 345–356.

[60] W. Xu, Y. Qi, and D. Evans, "Automatically evading classifiers," in *Proceedings of the 2016 Network and Distributed Systems Symposium*, 2016.

[61] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 1322–1333.

[62] X. Wu, M. Fredrikson, S. Jha, and J. F. Naughton, "A methodology for formalizing model-inversion attacks," in *Proceedings of the 2016 IEEE Computer Security Foundations Symposium (CSF)*. IEEE, 2016, pp. 355–370.

[63] M. Brückner and T. Scheffer, "Stackelberg games for adversarial prediction problems," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 547–555.

[64] M. Brückner, C. Kanzow, and T. Scheffer, "Static prediction games for adversarial learning problems," *Journal of Machine Learning Research*, vol. 13, no. Sep, pp. 2617–2654, 2012.

[65] B. Biggio, G. Fumera, and F. Roli, "Design of robust classifiers for adversarial environments," in *Proceedings of the 2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2011, pp. 977–982.

[66] R. N. Rodrigues, L. L. Ling, and V. Govindaraju, "Robustness of multimodal biometric fusion methods against spoof attacks," *Journal of Visual Languages & Computing*, vol. 20, no. 3, pp. 169–179, 2009.

[67] B. Biggio, G. Fumera, and F. Roli, "Multiple classifier systems for adversarial classification tasks," in *Proceedings of the International Workshop on Multiple Classifier Systems*. Springer, 2009, pp. 132–141.

[68] ——, "Multiple classifier systems for robust classifier design in adversarial environments," *International Journal of Machine Learning and Cybernetics*, vol. 1, no. 1-4, pp. 27–41, 2010.

[69] ——, "Multiple classifier systems under attack," in *Proceedings of the International Workshop on Multiple Classifier Systems*. Springer, 2010, pp. 74–83.

[70] A. Globerson and S. Roweis, "Nightmare at test time: robust learning by feature deletion," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 353–360.

[71] A. Kołcz and C. H. Teo, "Feature weighting for improved classifier robustness," in *Proceedings of the 6th Conference on Email and Anti-Spam*, 2009.

[72] A. Kantchelian, S. Afroz, L. Huang, A. C. Islam, B. Miller, M. C. Tschantz, R. Greenstadt, A. D. Joseph, and J. Tygar, "Approaches to adversarial drift," in *Proceedings of the 2013 ACM workshop on Artificial intelligence and security*. ACM, 2013, pp. 99–110.

[73] B. Liang, H. Li, M. Su, P. Bian, X. Li, and W. Shi, "Deep text classification can be fooled," *arXiv preprint arXiv:1704.08006*, 2017.

[74] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial examples for malware detection," in *European Symposium on Research in Computer Security*. Springer, 2017, pp. 62–79.

[75] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.

[76] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 1528–1540.

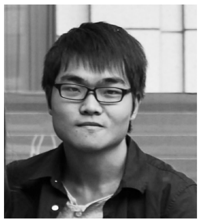
[77] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proceedings of the 2017 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2017.

[78] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *Proceedings of the Fifth International Conference on Learning Representations*, 2017.

[79] Q. Wang, W. Guo, K. Zhang, A. G. Ororbia II, X. Xing, C. L. Giles, and X. Liu, "Learning adversary-resistant deep neural networks," *arXiv preprint arXiv:1612.01401*, 2016.

[80] Q. Wang, W. Guo, K. Zhang, A. G. Ororbia II, X. Xing, X. Liu, and C. L. Giles, "Adversary resistant deep neural networks with an application to malware detection," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 1145–1153.

Bin Liang received the Ph.D. degree in Computer Science from Institute of Software, Chinese Academy of Sciences. He is currently a professor at School of Information, Renmin University of China. His research interests focus on program analysis, vulnerability detection and Web security.



Hongcheng Li received the B.S. degree in Information Security from Renmin University of China. He is currently working towards the M.S. degree in Information Security at School of Information, Renmin University of China. His research interests focus on machine learning and security.

Miaoqiang Su received the M.S. degree in Information Security from Renmin University of China. His research interests focus on machine learning.

Xirong Li received the Ph.D. degree in Computer Science from the University of Amsterdam. He is currently an Associate Professor at School of Information, Renmin University of China. His research interests include multimedia tagging, categorization, and retrieval.

Wenchang Shi received the Ph.D. degree in Computer Science from Institute of Software, Chinese Academy of Sciences. He is currently a professor at School of Information, Renmin University of China. His research interests focus on information security, trusted computing, cloud computing, computer forensics, and operating systems.

Xiaofeng Wang received the Ph.D. degree in computer engineering from Carnegie Mellon University in 2004. Since 2010, he has been the acting Director of the Security Informatics Program at Indiana University Bloomington, where he is currently a Professor with the School of Informatics. His research interests include cloud and mobile security, and data and health informatics security.