TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 02/13 pp800–812
DOI: 10.26599/TST.2020.9010038
Volume 26, Number 6, December 2021

A Cascade Model-Aware Generative Adversarial Example Detection Method

Keji Han, Yun Li*, and Bin Xia

Abstract: Deep Neural Networks (DNNs) are demonstrated to be vulnerable to adversarial examples, which are elaborately crafted to fool learning models. Since the accuracy and robustness of DNNs are at odds for the adversarial training method, the adversarial example detection algorithms check whether the specific example is adversarial, which is promising to solve the issue of the adversarial example. However, among the existing methods, model-aware detection methods do not generalize well, while the detection accuracies of the generative-based methods are lower compared to the model-aware methods. In this paper, we propose a cascade model-aware generative adversarial example detection method, namely CMAG. CMAG consists of two first-order reconstructors and a second-order reconstructor, which can illustrate what the model sees to the human by reconstructing the logit and feature maps of the last convolution layer. Experimental results demonstrate that our method is effective and is more interpretable compared to some state-of-the-art methods.

Key words: information security; Deep Neural Network (DNN); adversarial example detection

1 Introduction

Deep Neural Networks (DNNs)[1–3] become increasingly security-sensitive, even they have achieved excellent performance in many machine learning tasks, such as Computer Vision (CV)[3,4], Neural Language Processing (NLP)[5,6], and Speech Recognition (SR)[7]. DNNs are demonstrated to be vulnerable to adversarial examples[8–11]. Adversarial examples are crafted by the adversary to ruin the performance of the target model with specific attack algorithms[12,13]. In detail, the adversarial example is generated by adding adversarial perturbation into the legitimate example. According to the knowledge of the adversary, attack algorithms fall into two categories: white-box attack and black-box attack[9]. In the white-box attack scenario, the adversary has perfect knowledge of the target model, including architecture, parameters, and training set. As to the black-box attack, the adversary only has limited knowledge compared to the white-box scenario.

According to our knowledge, the existing attack methods fall into three categories, namely gradient-based, optimization-based, and gradient-free methods, according to the process to explore adversarial perturbation. Gradient-based attacks employ gradient information to craft adversarial examples: Fast Gradient Sign Method (FGSM)[14] crafts adversarial example with the sign of the gradient with respect to the ground-truth label; least-likely attack[15] adds the sign of the gradient with respect to the smallest probability label. Gradient-based attack methods can be formulated as

$$\boldsymbol{x}_{\text{adv}} = \boldsymbol{x} + \lambda \cdot G(\nabla_{\boldsymbol{x}} \ell(f(\boldsymbol{x};\boldsymbol{\theta}), \hat{y}) \qquad (1)$$

where $\boldsymbol{x}_{\text{adv}}$ is a crafted adversarial example, $\boldsymbol{x}$ is an example, and $\hat{y}$ is a label. $f(\cdot)$ is the target (victim) model with parameter set $\boldsymbol{\theta}$. $\nabla_{\boldsymbol{x}} \ell(f(\boldsymbol{x};\boldsymbol{\theta}), \hat{y})$ is the gradient of $\boldsymbol{x}$. $G(\cdot)$ is the gradient map function, such as sign function and unit function. λ is a hyperparameter to control the attack intensity.

Optimization-based methods formulate the process to explore adversarial perturbation into an optimization as follows:

• Keji Han, Yun Li, and Bin Xia are with the School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210046, China. E-mail: 1016041119@njupt.edu.cn; liyun@njupt.edu.cn; bxia@njupt.edu.cn.

∗ To whom correspondence should be addressed.

Manuscript received: 2020-09-01; accepted: 2020-09-14

$$\min_{\boldsymbol{\delta}} f(\boldsymbol{x}) \neq f(\boldsymbol{x}+\boldsymbol{\delta}),$$
$$\text{s.t.} \quad \|\boldsymbol{\delta}\|_p \leqslant \epsilon \tag{2}$$

where $\boldsymbol{\delta}$ and ϵ denote adversarial perturbation and the upper bound of l_p norm of the adversarial perturbation, respectively. For instance, Calini & Wagner (CW) attack[16] explores the smallest adversarial perturbation with l_p norm constraint. Deepfool[17] minimizes the distance between the adversarial example and the target hyperplane.

Gradient-free attacks craft the adversarial examples by exploring the corresponding adversarial perturbation with specific searching strategy or module. One pixel attack[18] explores adversarial perturbation with Differential Evolution (DE) algorithm, which employs no gradient information of the corresponding legitimate examples. Adversarial Transformation Network (ATN)[19] directly maps the legitimate example as an adversarial example, the difference between the original example and mapping result can be viewed as the adversarial perturbation, by employing a transformation network. The attack methods are summarized in Table 1.

To mitigate the impact of adversarial example, couples of defense algorithms have been proposed. According to the aim of defender, current defense algorithms are divided into two categories: model-strengthening and adversarial example detection. Model-strengthening defense algorithms try to protect the gradient of the target model, while detection algorithms are introduced to make the target model keep off the adversarial example, without changing the model itself.

Model-strengthening methods try to make the target model itself robust against adversarial examples with retaining itself or some supplementary modules. The effectiveness of model-strengthening methods generally roots in gradient-confused[20]. Adversarial training methods[21–24] aim to make the target model loss of examples to be zeros, vanishing the gradient. Feature nullification methods[25] try to mask the original gradients. Wang et al.[26] added a non-differential module into the target model, while retaining the accuracy.

Detection methods for adversarial examples of DNNs fall into two categories: metric-based and additional-model-based. Metric-based detection methods[27,28] check whether the given example is an adversarial example by specific metric, such as mean variance and kernel density. Additional-model-based methods can be divided into two classes: discriminative-model-based and generative-model-based. Discriminative-model-based[29] methods train a (or more) classifier(s) to check whether the given example is adversarial, while generative-model-based methods[30–32] try to reconstruct the given example, then evaluate whether the example is adversarial by calculating the Reconstruction Error (RE) between the original and the corresponding reconstruction result. Wang et al.[29] trained couples of binary classifiers with outputs of different convolution layers as inputs to detect adversarial examples. Feature squeezing[32] reduces the color-bit depth of original example. If the distance between logits of the original example and the corresponding color-bit depth-reduced example exceeds the predefined threshold, the original example will be detected as an adversarial example. MagNet[30] trains few autoencoders to detect adversarial examples. In detail, if the reconstruction error, l_2 norm of the difference between the original and the corresponding reconstruction result, exceeds the predefined threshold, the original example is detected as an adversarial example. Defense Generative Adversarial Network (Defense GAN)[31] is an extension of MagNet, which employs GAN to reconstruct the given inputs. Defense GAN introduces optimization strategy to search more efficient, which is a low-dimensional vector to fed to the generator. The defense methods are summarized in Table 2.

In model-strengthening methods, the legitimate example and the corresponding adversarial example are at odds to update the parameters of DNNs, so model-strengthening method will degrade the performance of the target model on legitimate examples. Detection defense methods are promising to mitigate the odds between the robustness and accuracy of the target model by keeping the target model off the adversarial example. However, few detection algorithms focus on interpretability and are ineffective to some optimization-based attack. In this paper, we propose a Cascade

Table 1 Current attack methods.

Attack type	Method
Gradient-based	FGSM[14], least-likely attack[15]
Optimization-based	CW[16], Deepfool[17]
Gradient-free	DE[18], ATN[19]

Table 2 Current defense methods.

Defense type	Method
Model-strengthening method	Adversarial training[21–24], feature nullification[25]
Adversarial example detection	Metric-based[27,28], additional-model-based[30–32]

Model-Aware Generative adversarial examples detection method (CMAG), which employs Structural SIMilarity (SSIM) index[33] as additional training loss and detection criterion. Different from current generative model-based methods, CMAG consists of two first-order reconstructors and a second-order reconstructor. By cascading reconstruction, CMAG explains what the target model sees to the human. The experimental results demonstrate that it is more efficient to detect optimization-based adversarial examples compared to current generative model-based methods.

Our contributions are summarized as follows.

- We propose a vision-interpretable adversarial examples detection method, CMAG;
- We theoretically prove the effectiveness of our method;
- We provide a new perspective of the existence of the adversarial example.

The rest of the paper is organized as follows. In Section 2, some related definitions involved in this paper are introduced. CMAG is introduced in Section 3. Experiments are introduced in Section 4. Discussions are conducted in Section 5, while conclusions are drawn in Section 6.

2 Preliminary

Before the introduction of the proposed CMAG, some related definitions will be introduced in this section.

2.1 l_p norm

l_p norm is generally employed to generative model. $\boldsymbol{X}$ is the training set. For a specific example, $\boldsymbol{x}_k \in \boldsymbol{X}$, $|\boldsymbol{x}_k|$ is the cardinality of $\boldsymbol{x}$, $|\boldsymbol{x}_k| = N$. $x_k^i \in \boldsymbol{x}_k, i = 1, 2, \ldots, N$, $l_p(\boldsymbol{x}_k)$ is formulated as

$$l_p(\boldsymbol{x}_k) = \sqrt[p]{\sum_{i=1}^{N} |x_k^i|^p} \tag{3}$$

l_1 and l_2 are common l_p norm. The corresponding derivatives of $\boldsymbol{x}_k$ are formulated as follows:

$$\frac{\partial l_2(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_k} = \frac{\partial \boldsymbol{x}_k^{\mathrm{T}} \boldsymbol{x}_k}{\partial \boldsymbol{x}_k} = 2\boldsymbol{x}_k,$$
$$\frac{\partial l_1(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_k} = \operatorname{sign}(\boldsymbol{x}_k) \tag{4}$$

2.2 Distillation

Distillation is proposed in Ref. [34]. DNN-based classifier commonly has a softmax layer in the image classification task. The softmax layer normalizes the output of the layer prior the softmax, $z(\boldsymbol{x})$, into a probability vector $\operatorname{softmax}(\boldsymbol{x})$, i.e., the logit. Each component $\operatorname{softmax}(\boldsymbol{x})_i$, $i \in 0, 1, \ldots, K-1$, denotes the probability that $\boldsymbol{x}$ belongs to the corresponding category, where K is the number of the classes. The distillation can be formulated as follows:

$$\operatorname{softmax}(\boldsymbol{x})_i = \frac{\mathrm{e}^{z(x)_i/T}}{\sum_{j=0}^{K-1} \mathrm{e}^{z(x)_j/T}}, \ i \in 0, 1, \ldots, K-1, T > 0 \tag{5}$$

where T is introduced to adjust the proportion of probabilities of the classes. For instance, when $T \to \infty$, $\operatorname{softmax}(\boldsymbol{x})_i$ is approximate $1/K$ for all $i \in 0, 1, \ldots, K-1$. Conversely, when $T \to 0$, the biggest element of $\operatorname{softmax}(\boldsymbol{x})$ is close to 1, while the rest elements are approximate to 0.

2.3 SSIM

l_p norm is always employed to evaluate the similarity between the legitimate example and the corresponding adversarial example. However, it may cause a difference between recognition results of DNNs and human, since l_p norm is not a human-vision consistent metric. So to get good interpretability, it is wise to introduce the human-sensitive criterion as the loss function. Zhao et al.[35] speculated that the human is sensitive to structural patterns change in an image. Zhou et al.[33] introduced SSIM to simulate the human-sensitive similarity between two images.

For a given example $\boldsymbol{x}_k$, the mean intensity is calculated as

$$\mu_{x_k} = \frac{1}{N} \sum_{i=1}^{N} x_k^i \tag{6}$$

where x_k^i $(i = 0, \ldots, N-1)$ is the i-th pixel of $\boldsymbol{x}_k$ and N is the number of the pixels. The standard deviation and the square root of the variance are applied to estimate the signal contrast. Furthermore, the unbiased standard deviation can be formulated as

$$\sigma_{x_k} = \frac{1}{N-1} \left(\sum_{i=1}^{N} (x_k^i - \mu_{x_k}) \right)^{1/2} \tag{7}$$

Moreover, for two examples, x_1^i and x_2^i, the covariance $\sigma_{x_1 x_2}$ can be estimated as

$$\sigma_{x_1 x_2} = \frac{1}{N-1} \sum_{i=1}^{N} (x_1^i - \mu_{x_1})(x_2^i - \mu_{x_2}) \tag{8}$$

Given two images $\boldsymbol{x}_1$and $\boldsymbol{x}_2$, the luminance $L(\boldsymbol{x}_1, \boldsymbol{x}_2)$, contrast $C(\boldsymbol{x}_1, \boldsymbol{x}_2)$, and structural $S(\boldsymbol{x}_1, \boldsymbol{x}_2)$ are defined as

$$L(\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{2\mu_{x_1}\mu_{x_2} + c_1}{\mu_{x_1}^2 + \mu_{x_2}^2 + c_1},$$

$$C(\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{2\sigma_{x_1}\sigma_{x_2} + c_2}{\sigma_{x_1}^2 + \sigma_{x_2}^2 + c_2},$$
$$S(\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{2\sigma_{x_1 x_2} + c_3}{\sigma_{x_1}\sigma_{x_2} + c_3} \quad (9)$$

where c_1, c_2, and c_3 are constants, introduced for the reason that $\mu_{x_1}^2 + \mu_{x_2}^2$ is commonly close to zero. From the definitions above, it can be speculated that luminance is the function of the mean, while the contrast depends on variance.

The SSIM index between examples $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is

$$\mathrm{SSIM}(\boldsymbol{x}_1, \boldsymbol{x}_2) = [L(\boldsymbol{x}_1, \boldsymbol{x}_2)]^{\alpha} \times [C(\boldsymbol{x}_1, \boldsymbol{x}_2)]^{\beta} \times [S(\boldsymbol{x}_1, \boldsymbol{x}_2)]^{\gamma} \quad (10)$$

where $\alpha > 0$, $\beta > 0$, and $\gamma > 0$ are super meters employed to adjust the relative importance of the three components. In this paper, we set $\alpha = \beta = \gamma = 1$.

According the definition of $L(\cdot)$, $C(\cdot)$, and $S(\cdot)$, it can be known that $L(\cdot), C(\cdot)$, and $S(\cdot) \in (0, 1]$, for $\mu_{x_1}^2 + \mu_{x_2}^2 \geqslant 2\mu_{x_1}\mu_{x_2}$, $\boldsymbol{x}_1^i \in [0, 1]$, $\boldsymbol{x}_2^i \in [0, 1]$, and $i = 0, \ldots, N-1$.

2.4 Adversarial training

Since adversarial examples can ruin the performance of target DNN[10, 13], the defender can retrain the DNN to with adversarial example to improve the robustness, which is the basic idea of the adversarial training. Furthermore, we can formulate the retraining process of DNN as follows:

$$\min_{\theta} \alpha \sum_{\boldsymbol{x} \in \boldsymbol{X}} \ell(f(\boldsymbol{x}; \boldsymbol{\theta}), y) + (1 - \alpha) \sum_{\boldsymbol{x} \in \boldsymbol{X}_{\mathrm{adv}}} \ell(f(\boldsymbol{x}; \boldsymbol{\theta}), y) \quad (11)$$

where $\boldsymbol{x}$ is an example, while y is its corresponding label. If $\boldsymbol{x}$ is an adversarial example, y is the label of its corresponding legitimate example. $\boldsymbol{X}$ and $\boldsymbol{X}_{\mathrm{adv}}$ are the legitimate example set and adversarial example set, respectively. α is a hyperparameter to balance the importance of legitimate examples and adversarial ones. $\ell(\cdot)$ is the loss function, commonly being the cross-entropy for the classification task. As shown in Eq. (11), adversarial training can be viewed as an extension of regulation, which also demonstrates the effectiveness of the adversarial training to improve the robustness of the target model. However, Eq. (11) also demonstrates that the adversarial example competes with the legitimate example to update parameters of the target model. That is the reason we focus on the adversarial example detection methods.

3 CMAG

Adversarial example detection is the primary task of mitigating the odds between the accuracy and robustness of the target model. As the definition in Ref. [12], adversarial examples are examples that crafted by the adversary to fool DNNs, while being imperceptible to the human. To detect the adversarial example, it is wise to known the feature representation of the input in the target model. So we conduct experiments to reconstruct the outputs of the last feature extraction module and the last fully connected layer. Here, we denote the victim model as $f(\cdot)$, and the target model of layers from the input layer to the last feature extraction layer as $f_{\mathrm{fm}}(\cdot)$. DNNs are generally trained with l_p norm or cross-entropy loss, which are insensitive and low-interpretability to humans. So it is vital to explore the human-sensitive loss to improve the interpretability of DNNs. For instance, in computer vision task, luminance, contrast, structural, etc., are demonstrated to be human-sensitive[35], which should be introduced to train DNNs. Since SSIM is the function of the luminance, contrast, and structural measures related, it can be employed as loss function to train DNNs to improve the interpretability.

To understand why adversarial examples can attack DNNs, it is necessary to visualize the feature representation of adversarial examples in DNNs, which is the motivation of our detection method. Since the model-aware method uses the feature maps of the target model, it can achieve higher detection accuracy than the model-agnostic method. So we reconstruct the original input of the target model with the feature maps and the final probability vector of the target model. The workflow of CMAG, which consists of the SSIM-based detector, is presented in Fig. 1. Modules in the blue dash box are the target model, taking the classification model as an instance. The specific process is depicted as follows:

(1) Feature-Map Reconstructor (FMR) reconstructs the original input with the output of the feature extraction module, and Logit Reconstructor (LR) reconstructs the original input with the output of the softmax layer, i.e., the logit;

(2) Employ the output of the FMR as the attention of the output of LR by calculating their Hardmard product $h(\boldsymbol{x})$ as the input of the Global Reconstructor (GR);

(3) GR reconstructs the original input again with $h(\boldsymbol{x})$ as the input, marking its output as $\boldsymbol{x}'$. GR is the module

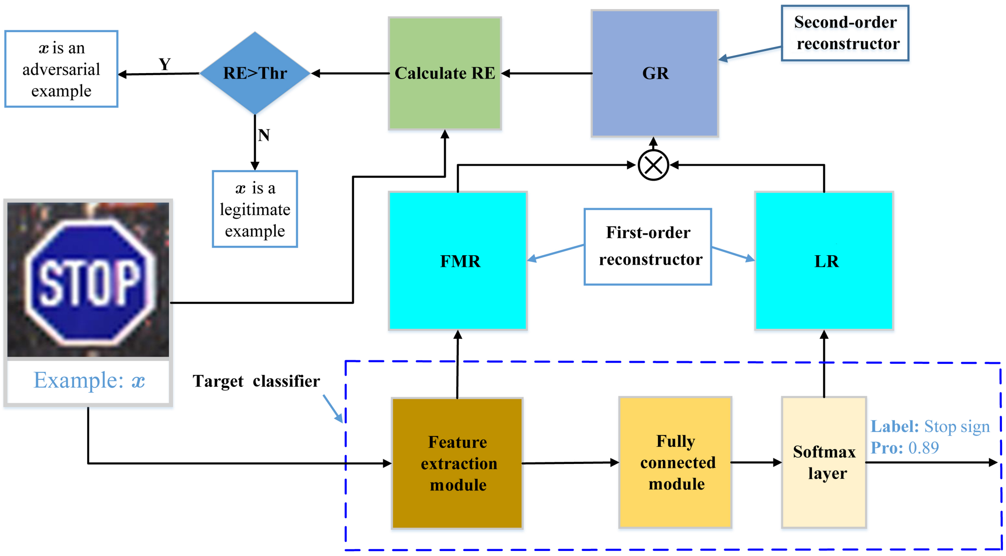

x is an adversarial example
Y
RE>Thr
N
x is a legitimate example
Calculate RE
GR
Second-order reconstructor
FMR
First-order reconstructor
LR
STOP
Example: x
Target classifier
Feature extraction module
Fully connected module
Softmax layer
Label: Stop sign
Pro: 0.89

Fig. 1 Workflow of CMAG in the deployment phase. CMAG includes two first-order reconstruct, named FMR and LR, and a second-order reconstructor, named GR. If the Reconstruct Error (RE) of x with respect to GR exceeds the given threshold Thr, it will be detected as an adversarial example.

to interpret what the target model sees from the original input;

(4) The detector module evaluates the similarity (or distance) between the reconstruction result and the original input to check whether the original input is an adversarial example.

The loss of FMR can be formulated as follows:

$$\ell_{\text{fmr}} = \mu_{\text{fmr}} \|\text{FMR}(f_{\text{fm}}(\boldsymbol{x})) - \boldsymbol{x}\|_2^2 + (1-\mu_{\text{fmr}})\text{SSIM}(\text{FMR}(f_{\text{fm}}(\boldsymbol{x})), \boldsymbol{x}) \quad (12)$$

The loss of LR is

$$\ell_{\text{lr}} = \mu_{\text{lr}} \|\text{LR}(\text{softmax}(f(\boldsymbol{x}))) - \boldsymbol{x}\|_2^2 + (1-\mu_{\text{lr}})\text{SSIM}(\text{LR}(\text{softmax}(f(\boldsymbol{x}))), \boldsymbol{x}) \quad (13)$$

The loss of the GR is

$$\ell_{\text{gr}} = \mu_{\text{gr}} \|\boldsymbol{x}' - \boldsymbol{x}\|_2^2 + (1-\mu_{\text{gr}})\text{SSIM}(\boldsymbol{x}', \boldsymbol{x}) \quad (14)$$

where μ_{fmr}, μ_{lr}, and μ_{gr} are hyperparameters to balance the importance between l_p loss and SSIM loss. FMR$(\cdot)$ and LR$(\cdot)$ are the FMR and LR module functions, respectively. Furthermore, the detail training processes of GR are shown in Algorithm 1, where $\nabla_{\theta}\ell_{\text{gr}}$ is the partial derivate of ℓ_{gr} with respective to $\boldsymbol{\theta}$.

3.1 FMR

FMR is utilized to map outputs of $f_{\text{fm}}(\cdot)$ into the original feature space. The training loss function of FMR is shown in Eq. (12). The role of FMR is to reconstruct the original input with high-level feature representation, which contains semantic features. An adversarial example is quasi-imperceptible to humans, while it can fool the target model. According to this observation, we speculate that the low-level feature maps of the original example and the corresponding adversarial example should be similar, so it is hard to detect adversarial examples with low-level feature maps (or even the original example itself). Furthermore, compared to the output of $f(\cdot)$, the output of $f_{\text{fm}}(\cdot)$ is lower-level features. Therefore, it is useful to consider the final output of the target model as the augmented feature.

Algorithm 1 Training process of GR

```
Require: Legitimate examples sets $X$, pretrained FMR and LR; GR's parameter $\boldsymbol{\theta}$; training epochs $I$; and learning rate $\eta$.
for $i = 1$ to $I$ do
    for $\boldsymbol{x}$ in $X$ do
        $h(\boldsymbol{x}) \leftarrow \text{FMR}(f_{\text{fm}}(\boldsymbol{x})) \otimes \text{LR}(f(\boldsymbol{x}))$
        $\boldsymbol{x}' \leftarrow \text{GR}(h(\boldsymbol{x}))$
        Calculate $\ell_{\text{gr}}$ between $\boldsymbol{x}'$ and $\boldsymbol{x}$ according to Eq. (14))
        $\boldsymbol{g_\theta} \leftarrow \nabla_{\boldsymbol{\theta}}\ell_{\text{gr}}$
        $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \cdot \boldsymbol{g_\theta}$
    end for
end for
```

3.2 LR

FMR tries to learn the semantic region of the original example, while LR is designed to reconstruct the semantic features of the original example with the logit. The loss function of LR is shown in Eq. (13). According to the definition of the adversarial example, the logits of the legitimate example and its corresponding adversarial example are different. In detail, the biggest elements for logits of the legitimate and adversarial example are different. In some cases, the difference is so subtle that it is difficult to correctly reconstruct the original input with the logit. Fortunately, the distillation can be adopted to mitigate this issue, which can endow the logits of adversarial examples and the corresponding legitimate with similar statistical characteristics. According to Eq. (5), smaller T ($T > 0$) increases the proportion of the max component, while decreases proportions of other components. For instance, the target model, with the softmax layer and $T = 1$, does not confirm which category the adversarial example belongs to, if the top-2 elements of the logit are 0.491 and 0.490. When T is reset as $1/1000$, the values of top-2 approximately become 1 and 0. However, the probability vector's dimension is so small that it is difficult to reconstruct the original example with only the logit. So we introduce GR, which takes the combination of outputs of FMR and LR as input.

3.3 GR

GR is introduced to interpret what the target mode realizes about the original input, taking the Hardmard product of the outputs of FMR and LR as input. Since FMR and LR focus on different level feature representations, combining them as the input of GR helps to improve the quality of reconstruction example. The experimental results also demonstrate that GR achieves more excellent reconstruction performance.

3.4 Adversarial example detector

Adversarial example detector checks whether the given example is an adversarial example, according to the outputs of the GR. We apply variant criteria, such as SSIM and l_2, to our detctor to evaluate differences between $\boldsymbol{x}$ (the original example) and $\boldsymbol{x}'$ (the output of GR). The detection threshold searching module is shown in Algorithm 2. According to Eqs. (3) and (10), if the reconstruction error exceeds any threshold determined in Algorithm 2, the example will be detected as an adversarial example.

Algorithm 2 SSIM-based adversarial example detection algorithm

```
Require: Training set $X$; $f(\cdot)$; $f_{\text{fm}}(\cdot)$; $\text{Thr}_{\text{SSIM}}$ and $\text{Thr}_{l_2}$ (thresholds of reconstruction errors for the SSIM and $l_2$ detectors, respectively); output of GR $\boldsymbol{x}'$
  Initialize $\text{Thr}_{\text{SSIM}} \leftarrow 1$, $\text{Thr}_{l_2} \leftarrow 0$
  for $\boldsymbol{x}$ in $X$: do
    Reconstruct the example with FMR and LR, denoted as $\text{FMR}(f_{\text{fm}}(\boldsymbol{x}))$ and $\text{LR}(\text{softmax}(f(\boldsymbol{x})))$;
    Calculate the Hadmard product between outputs of FMR and LR, $h(\boldsymbol{x}) = \text{FMR}(f_{\text{fm}}(\boldsymbol{x})) \otimes \text{LR}(f(\boldsymbol{x}))$;
    Reconstruct the example with GR, taking $h(\boldsymbol{x})$ as input, marked as $\boldsymbol{x}'$;
    Calculate $\text{SSIM}(\boldsymbol{x}, \boldsymbol{x}')$ and $l_2(\boldsymbol{x}, \boldsymbol{x}')$ according to Eqs. (10) and (3), respectively.
    if $\text{SSIM}(\boldsymbol{x}, \boldsymbol{x}') < \text{Thr}_{\text{SSIM}}$ then
      $\text{Thr}_{\text{SSIM}} = \text{SSIM}(\boldsymbol{x}, \boldsymbol{x}')$;
    end if
    if $l_2(\boldsymbol{x}, \boldsymbol{x}') > \text{Thr}_{l_2}$ then
      $\text{Thr}_{l_2} = l_2(\boldsymbol{x}, \boldsymbol{x}')$;
    end if
  end for
  return $\text{Thr}_{\text{SSIM}}$ and $\text{Thr}_{l_2}$
```

4 Experiment

4.1 Experiment setting

Three datasets, MNIST[36], Fashion-MNIST (FMNIST)[37], and GTSRB[38] are employed to evaluate the performance of defense methods involved in this paper. MNIST consists of a training set, including 60 000 grayscale hand-written digits images with size 28 pixel × 28 pixel of 10 classes, and a testing set with 10 000 examples. FMNIST is an extension of MNIST. In detail, FMNIST also consists of 10 categories, named T-shirt, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, and Ankle boot. Moreover, the number of examples and the size of the example are the same with MNIST. As to GTSRB, since examples of it with variant size, we scale them into uniform size 32 pixel × 32 pixel. The train set of GTSRB consists of 39 209 color image examples of 43 categories, while the testing set consists of 12 630 examples.

We set $\mu_{\text{fmr}} = \mu_{\text{lr}} = \mu_{\text{gr}} = 0.5$ in Eqs. (12)–(14), which is consistent with the setting of Zhao et al.[35]. FMR, LR, and GR are trained with Adam[39] optimizer, learning rate is 10^{-3}. In training phase, the batch size is set 2000, while in testing phase batch size is 200 for three datasets. FGSM and CW_{l_2} attack methods are adopted to evaluate the robustness of detection algorithms involved in this paper, and CW_{l_2} is the CW attack with l_2 norm. Moreover, to improve the detection difficulty, attack intensities of CW_{l_2} are set to be small.

4.2 Detection accuracy

In this section, we do experiments to compare the detection accuracies of our method with some state-of-the-art generative detection methods, under FGSM and CW_{l_2} attacks with different attack intensities.

The details of our method mentioned in Tables 3–5 are shown in Algorithm 2. As shown in Tables 3–5, our method achieves best detection performance, compared to MagNet and Defense GAN. According to the Tables 3–5, we note that all generative detection methods are efficient to detect FGSM adversarial example, while MagNet and Defense GAN are poor at detecting CW adversarial examples, since the attack intensity is small. To demonstrate the difference between FGSM and CW adversarial examples, we show two types of adversarial examples in Fig. 2.

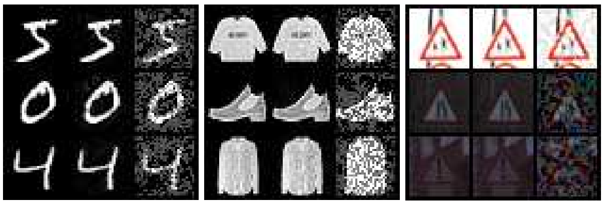

(a) MNIST (b) FMNIST (c) GTSRB

Fig. 2 Legitimate and the corresponding CW and FGSM adversarial examples of MNIST, FMNIST, and GTSRB. In each subfigure, the first column is the legitimate example and the second and third columns are the CW and FGSM adversarial examples, respectively. Moreover, the attack intensity of FGSM is 0.3, and the attack intensity of CW is 19.

As shown in Fig. 2, different from FGSM adversarial examples, CW adversarial examples of three datasets are almost imperceptible to human, since the attack intensity is set small. As mentioned in Section 3, the adversarial example and its corresponding legitimate are almost the same, while logits corresponding them are much more different. The effectiveness of our method to detect CW adversarial example roots in that it can measure misalignment between variant-level features of the target model. If an example is adversarial, high-level features (semantic features) misalign with the corresponding low-level features (original pixel features). Our method maps the high-level feature into the original feature space, then compares it with the original example. Different from the existing methods, our method is more efficient when the ℓ_p norm of the adversarial perturbation is smaller.

Table 3 Comparison of detection accuracies of our method with MagNet and Defense GAN, under different attack intensities on MNIST, where λ and confidence represent the attack intensities of FGSM and CW_{l_2}, respectively.

Detection method	Attack intensity					
	λ			Confidence		
	0.1	0.2	0.3	9	19	29
MagNet	1.00	1.00	1.00	1.00	1.00	0.21
Defense GAN	1.00	1.00	1.00	0.00	0.02	0.05
Our method	1.00	1.00	1.00	1.00	1.00	1.00

Table 4 Comparison of detection accuracies of our method with MagNet and Defense GAN, under FGSM and CW_{l_2} with different attack intensities on FMNIST.

Detection method	Attack intensity					
	λ			Confidence		
	0.1	0.2	0.3	9	19	29
MagNet	1.00	1.00	1.00	0.00	0.00	0.23
Defense GAN	1.00	1.00	1.00	0.00	0.00	0.00
Our method	1.00	1.00	1.00	1.00	1.00	1.00

Table 5 Comparison of detection accuracies of our method with MagNet and Defense GAN, under FGSM and CW_{l_2} with different attack intensities on GTSRB.

Detection method	Attack intensity					
	λ			Confidence		
	0.1	0.2	0.3	9	19	29
MagNet	1.00	1.00	1.00	0.00	0.00	0.00
Defense GAN	1.00	1.00	1.00	0.00	0.00	0.00
Our method	1.00	1.00	1.00	1.00	1.00	1.00

To further illustrate the effectiveness of our method, we show the average SSIM similarity between the adversarial example and its corresponding reconstruction result for FGSM and CW_{l_2} in Table 6. Threshold represents the average SSIM similarity between legitimate example and its corresponding reconstruction result. As shown in Table 6, since the threshold is much bigger than the SSIM similarity between the adversarial example and its corresponding reconstruction result, we can speculate that our method makes a clear distinction between the legitimate and adversarial examples. Moreover, SSIM similarity between the adversarial example and the corresponding

Table 6 Average SSIM similarities between reconstruction results and corresponding adversarial examples of MNIST, FMNIST, and GTSRB. Threshold represents the smallest SSIM similarity between the legitimate inputs and corresponding reconstruction results.

Dataset	Attack intensity						Threshold
	λ			Confidence			
	0.1	0.2	0.3	9	19	29	
MNIST	0.442	0.442	0.442	0.654	0.532	0.442	0.824
FMNIST	0.145	0.101	0.074	0.400	0.375	0.331	0.763
GTSRB	0.362	0.185	0.109	0.748	0.712	0.674	0.757

reconstruction result decreases with the attack intensity increasing. We can conclude that higher attack intensity of adversarial examples, the easier detection. Furthermore, factors that impact the detection accuracy will be discussed in the following section.

4.3 Theoretical analysis of the effectiveness of CMAG

According to the Ref. [35], we know

$$\frac{1}{\mathrm{SSIM}(\boldsymbol{x}_1,\boldsymbol{x}_2)} = 1 + \frac{\mathrm{MSE}(\boldsymbol{x}_1,\boldsymbol{x}_2)}{2\sigma_{x_1x_2} + c_2} \tag{15}$$

Then we can get

$$\|\boldsymbol{x}_1-\boldsymbol{x}_2\|_2 = \sqrt{N} \times \frac{1-\mathrm{SSIM}(\boldsymbol{x}_1,\boldsymbol{x}_2)}{\mathrm{SSIM}(\boldsymbol{x}_1,\boldsymbol{x}_2)}(2\sigma_{x_1x_2} + c_2) \tag{16}$$

So we speculate that the detectabilities of the adversarial example under SSIM metric and l_p-norm metric are equivalent. To be simple, we just prove the detectability of the adversarial example under l_2 metric.

Theorem 1 Since modules of the CMAG are only trained with legitimate examples, we assume that $\|\phi(f(\boldsymbol{x}))-\boldsymbol{x}\|_2 \leqslant e_1$, $\|\phi(f(\boldsymbol{\epsilon}))-\boldsymbol{\epsilon}\|_2 \geqslant e_2$, and $e_2 > 2e_1$, $\phi(\cdot)$ is the reconstruction model. $\boldsymbol{\Omega}$ and $\boldsymbol{x}$ represent the example space and an example of $\boldsymbol{\Omega}$, $\forall \boldsymbol{x} \in \boldsymbol{\Omega}$, and $\boldsymbol{\epsilon}$ is its corresponding adversarial perturbation. e_1 and e_2 are constants. Then $\|\phi(f(\boldsymbol{x}+\boldsymbol{\epsilon}))-(\boldsymbol{x}+\boldsymbol{\epsilon})\|_2 \geqslant e_1$.

Proof

$$\begin{aligned}&\|\phi(f(\boldsymbol{x}+\boldsymbol{\epsilon}))-(\boldsymbol{x}+\boldsymbol{\epsilon})\|_2 \approx\\ &\|\phi(f(\boldsymbol{x})-\boldsymbol{x}+\phi(f(\boldsymbol{\epsilon}))-\boldsymbol{\epsilon}\|_2 \geqslant\\ &\|\phi(f(\boldsymbol{\epsilon}))-\boldsymbol{\epsilon}\|_2 - \|\phi(f(\boldsymbol{x}))-\boldsymbol{x}\|_2 \geqslant\\ &e_2 - e_1 > e_1\end{aligned} \tag{17}$$

According to Theorem 1, we know that $\forall \boldsymbol{x} \in \boldsymbol{\Omega}$, if $\boldsymbol{x}$ is a legitimate example, the reconstruction error is less than or equal to e_1, while if $\boldsymbol{x}$ is an adversarial example, the reconstruction error is bigger than e_1.

To test the validity of the hypothesis, we conduct the experiment that compares the reconstruction errors of the legitimate example and the corresponding adversarial perturbation, and the experimental results are shown in Table 7.

Table 7 l_2 reconstruction error of legitimate and adversarial examples on MNIST, FMNIST, and GTSRB.

Dataset	Legitimate	Attack intensity					
		λ			Confidence		
		0.1	0.2	0.3	9	19	29
MNIST	3.165	9.469	10.551	12.073	8.841	8.980	9.111
FMNIST	3.771	9.680	10.696	12.335	9.352	9.850	10.293
GTSRB	6.213	14.846	18.355	22.185	12.535	12.682	12.687

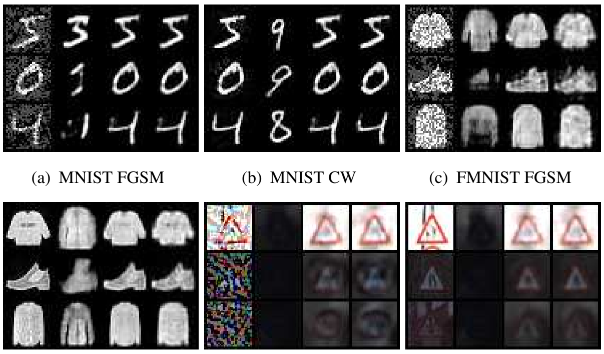
5 Discussion

In this section, some additional experiments are conducted to explain the details of the proposed method.

(1) Why outputs of $f_{\mathrm{fm}}(\cdot)$ and $f(\cdot)$ are chosen to reconstruct the given example?

The adversarial example can fool the DNNs while having little impact on human vision, which means that there is a distinct difference between human and computer vision. To improve the interpretability of the detection algorithm, it is wise to explore the target model from the given example. Low-level feature maps represent local features, such as gray values and edge features, while high-level features contain more semantic information[40]. The outputs of $f_{\mathrm{fm}}(\cdot)$ and $f(\cdot)$ as high-level features contain more semantic information that is different for the adversarial example and its corresponding legitimate. So our method employs outputs of $f_{\mathrm{fm}}(\cdot)$ and $f(\cdot)$ to reconstruct the given example.

We may also concern the ground-truth reconstruct results. MagNet and Defense GAN reconstruct the original example themselves without any information from the target model, which is different from our method. Additional experiments are conducted to show ground-truth reconstruction results of adversarial examples for our method, MagNet, and Defense GAN, and the experimental results are shown in Fig. 3. As shown in Fig. 3, the ground-truth reconstruction results of our method illustrate what the target model realizes for the given adversarial example, while reconstruction results of MagNet and Defense GAN are similar with original examples. The more significant the difference


(a) MNIST FGSM (b) MNIST CW (c) FMNIST FGSM

(d) FMNIST CW (e) GTSRB FGSM (f) GTSRB CW

Fig. 3 Comparison of ground-truth reconstruction results of our method, MagNet, and Defense GAN on adversarial examples of MNIST, FMNIST, and GTSRB. In each subfigure, from left to right, the original adversarial examples are in the first column, its corresponding reconstruction examples of our method, MagNet, and Defense GAN are in columns 2–4.

between the original example and corresponding reconstruction result, the more likely the example is marked as an adversarial example. Since the adversarial example is crafted to fool the target model, the feedback of the target model is helpful to detect adversarial examples, which is the difference between our method and other generative model-based methods. As shown in Fig. 3b, our method can even output an example belong to the adversarial category.

(2) What impacts the detection accuracy of the generative method?

The effectiveness of the generative method roots in the difference between reconstruction errors of the legitimate example and adversarial example. In detail, the adversarial example achieves higher reconstruction error compared to the legitimate example. We will introduce two factors which moderately impact the reconstruction error in the following paragraphs.

Fitting the ground-truth distribution of legitimate examples is a vital factor that impacts the detection accuracy of the given generative method. To better fit the ground-truth distribution, the employed generative model should be endowed suitable complexity to get enough representation ability, while being intractable to train. Furthermore, the employed generative model's performance is determined by some involving factors, such as training data, training epochs, and loss function.

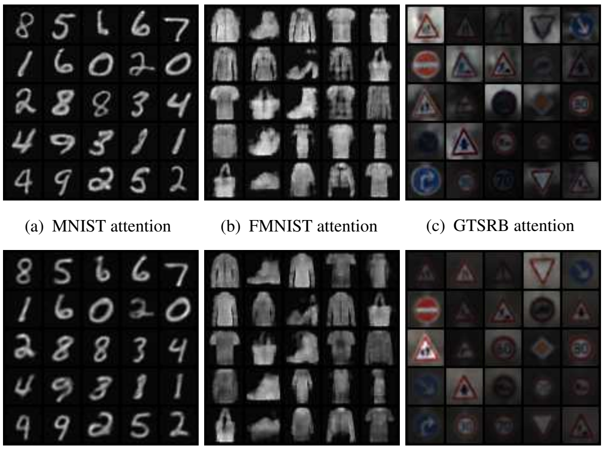
Metric of the reconstruction error is another factor that impacts detection accuracy. We should select a sensitive metric to measure the difference between the original input and the corresponding reconstruction result. Statistic characteristics and l_p norm are commonly employed to evaluate the difference. However, metric mentioned above is lack of interpretability for human. In Ref. [35], it is demonstrated that the human is not sensitive to l_2 perturbation, while DNNs do. So it is vital to design more interpretable and sensitive metric for adversarial examples detection.

(3) Why attention is adopted to combine outputs of $f_{\text{fm}}(\cdot)$ and $f(\cdot)$?

Attention[6] is commonly applied to NLP task to make the model get long-time memory. In our detection method, we speculate the output of the FMR can be employed to help LR to reconstruct the given example, since outputs of LR and FMR are highly relative. Moreover, feature maps derive logit, which is similar to the relationship between the corpus and the semantic. Furthermore, we conduct experiments to reconstruct the given example with different combinations of outputs of LR and FMR, and the results are shown in Fig. 4. As shown in Fig. 4, the combination of outputs of LR and FMR endows GR with better reconstruction performance, compared to the Weighted Average (WA). We speculate that attention keeps the integrity of the information outputs of LR and FMR. Furthermore, to explore the functions of modules in our method, ablation experiments are conducted. In detail, we conduct experiments to compare the reconstruction results of LR, FMR, and GR on MNIST and GTSRB, and the results are shown in Figs. 5 and 6, respectively. As shown in Figs. 5 and 6, descending order of reconstruction results with respect to three generative modules (LR, FMR, and GR) is GR>FMR>LR. The order coincides with the scale of the information of inputs of three generative models, demonstrating that better reconstruction performance requires sufficient information as input.

(4) A new perspective on existence of the adversarial example

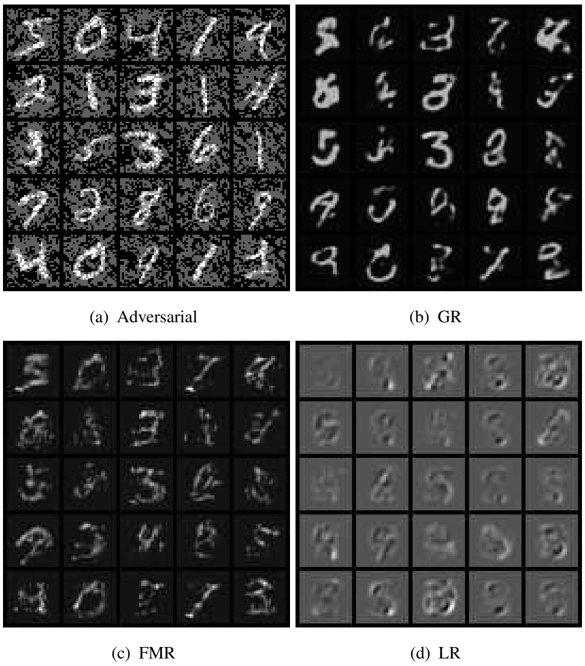
We speculate that it is the difference between human and computer visions that causes the existence of adversarial examples. In other words, it is hard to avoid the existence of the adversarial example. Shafahi et al.[41] speculated that the adversarial example is inevitable. Furthermore, Tsipras et al.[42] concluded that robustness may be at odds with accuracy, and the target model rarely achieves 100% accuracy on adversarial example. However, we can take measures to mitigate the



(a) MNIST attention (b) FMNIST attention (c) GTSRB attention

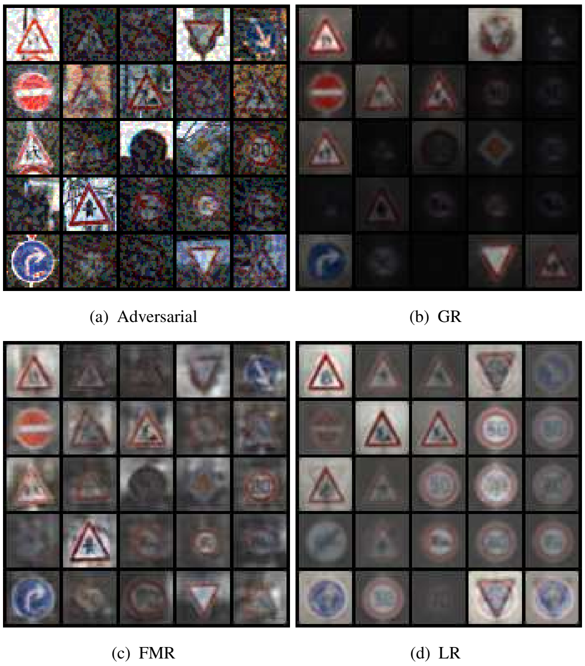
(d) MNIST WA (e) FMNIST WA (f) GTSRB WA

Fig. 4 Comparison of reconstruction results of GR with different combinations of outputs of FMR and LR on MNIST, FMNIST, and GTSRB. In detail, attention represents to calculate the dot product of outputs of FMR and LR, while the WA represents sum outputs of FMR and LR with specific normalized weight. Here we set the two weights to 0.5.


(a) Adversarial (b) GR

(c) FMR (d) LR

Fig. 5 Comparison of reconstruction performance of GR, FMR, and LR on adversarial examples of MNIST.


(a) Adversarial (b) GR

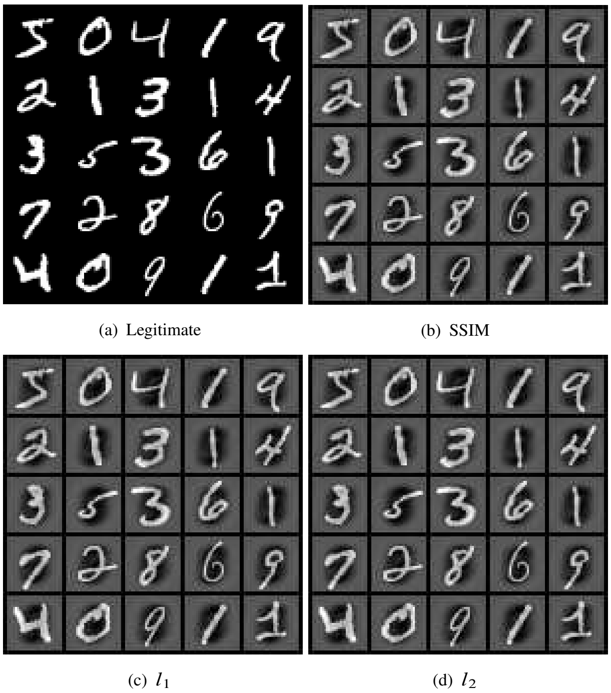
(c) FMR (d) LR

Fig. 6 Comparison of reconstruction performance of GR, FMR, and LR on adversarial examples of GTSRB.

impact of the adversarial example by forcing DNNs to be human-like. For instance, SSIM similarity corresponds with human vision. For an image pair, they are more similar for the human with the higher similarity of the SSIM. So we conduct experiments to compare the robustness of the generative model, autoencoder, which is trained with SSIM loss and l_2 norm loss. The results are shown in Figs. 7 and 8. According to Figs. 7 and 8, we note that the gradient of l_2 loss trained model is moderately correlated with semantic regions, while the gradient of the model trained with SSIM loss aligns well with the semantic regions (two models are trained with the same process except the loss function). That is to say, FGSM attack is unable to ruin an autoencoder trained with SSIM loss. Furthermore, we can conclude that generative models trained with SSIM loss are excellent at protecting its own gradients, compared the model trained with l_2 loss. Moreover, Elsayed et al.[43] added the retina layer to DNNs, making DNNs behave strikingly similar to humans. So we can design special architectures and losses to make DNNs behave like humans to improve the robustness of DNNs.

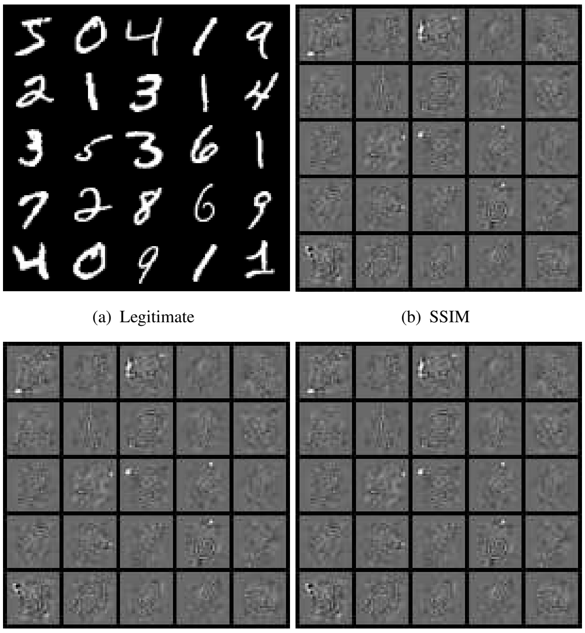
6 Conclusion

In this paper, we propose a cascade target model-aware generative adversarial example detection method that effectively detects high-quality adversarial examples and is more interpretable compared to current generative model-based detection methods. High-quality


(a) Legitimate (b) SSIM

(c) l_1 (d) l_2

Fig. 7 Comparison of legitimate examples and gradients of FGSM adversarial examples crafted with variant loss on MNIST. SSIM, l_1, and l_2 represent that the FGSM adversarial examples are crafted with SSIM, l_1, and l_2 loss, respectively. The victim autoencoder is trained with SSIM loss.



(a) Legitimate

(b) SSIM

(c) l_1

(d) l_2

Fig. 8 Comparison of legitimate examples and corresponding gradients of FGSM adversarial examples crafted with variant loss on MNIST. In detail, SSIM, l_1, and l_2 represent that the FGSM adversarial examples are crafted with SSIM, l_1, and l_2 loss, respectively. The victim autoencoder is trained with l_2 loss.

adversarial examples are adversarial examples within a small l_p neighborhood. By introducing the interpretable loss, SSIM, to our generative adversarial example detection, we provide a new perspective of the existence of the adversarial example, which highlights the way to explore more robust machine learning models for the computer vision task. In this paper, simple autoencoder is employed as the generative model, whose performance is not so reasonable for complicated dataset. In the future, more efficient generative model will be explored to improve the reconstruction performance of our method.

Acknowledgment

This work was partially supported by the National Natural Science Foundation of China (Nos. 61603197, 61772284, and 61876091).

References

[1] Y. Lecun, Y. Bengio, and G. Hinton, Deep learning, *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet classification with deep convolutional neural networks, in *Proc. of 26th Annual Conference on Neural Information Processing Systems 2012*, Lake Tahoe, NV, USA, 2012, pp. 1106–1114.

[3] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, Inception-v4, inception-resNet and the impact of residual connections on learning, in *Proc. of the Thirty-First Conference on Artificial Intelligence*, San Francisco, CA, USA, 2017, pp. 4278–4284.

[4] J. Cheng, Y. Li, J. Wang, L. Yu, and S. Wang, Exploiting effective facial patches for robust gender recognition, *Tsinghua Science and Technology*, vol. 24, no. 3, pp. 333–345, 2019.

[5] V. Yadav and S. Bethard, A survey on recent advances in named entity recognition from deep learning models, in *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, NM, USA, 2018, pp. 2145–2158.

[6] Y. Jiang, J. Cai, and K. Tu, Robust unsupervised discriminative dependency parsing, *Tsinghua Science and Technology*, vol. 25, no. 2, pp. 192–202, 2020.

[7] D. Jurafsky and J. H. Martin, *Speech and Language Processing*: *An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, 2nd edition, Prentice Hall Series in Artificial Intelligence. Upper Saddle River, NJ, USA: Prentice Hall, 2009.

[8] N. Papernot, P. D. McDaniel, A. Sinha, and M. P. Wellman, Towards the science of security and privacy in machine learning, arXiv preprint ar Xiv: 1611.03814, 2016.

[9] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, The security of machine learning, *Machine Learning*, vol. 81, no. 2, pp. 121–148, 2010.

[10] N. Papernot, P. D. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, The limitations of deep learning in adversarial settings, in *Proc. of European Symposium on Security and Privacy*, Saarbrücken, Germany, 2016, pp. 372–387.

[11] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, Intriguing properties of neural networks, arXiv preprint arXiv:1312.6199, 2013.

[12] N. Papernot, P. D. McDaniel, and I. J. Goodfellow, Transferability in machine learning: From phenomena to black-box attacks using adversarial samples, arXiv preprint ar Xiv: 1605.07277, 2016.

[13] F. Tramèr, N. Papernot, I. J. Goodfellow, D. Boneh, and P. D. McDaniel, The space of transferable adversarial examples, arXiv preprint ar Xiv: 1704.03453, 2017.

[14] I. J. Goodfellow, J. Shlens, and C. Szegedy, Explaining and harnessing adversarial examples, arXiv preprint arXiv: 1412.6572, 2014.

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

[16] N. Carlini and D. A. Wagner, Towards evaluating the robustness of neural networks, in *Proc. of 2017 IEEE Symposium on Security and Privacy*, San Jose, CA, USA, 2017, pp. 39–57.

[17] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, DeepFool:

A simple and accurate method to fool deep neural networks, in *Proc. of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, pp. 2574–2582.

[18] J. Su, D. V. Vargas, and K. Sakurai, One pixel attack for fooling deep neural networks, *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.

[19] S. Baluja and I. Fischer, Adversarial transformation networks: Learning to generate adversarial examples, arXiv preprint ar Xiv: 1703.09387, 2017.

[20] A. Athalye, N. Carlini, and D. A. Wagner, Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples, in *Proc. of the 35th International Conference on Machine Learning*, Stockholm, Sweden, 2018, pp. 274–283.

[21] F. Tramèr, A. Kurakin, N. Papernot, I. J. Goodfellow, D. Boneh, and P. D. McDaniel, Ensemble adversarial training: Attacks and defenses, in *Proc. of 6th International Conference on Learning Representations*, Vancouver, Canada, 2018, p. 582.

[22] Q. Cai, C. Liu, and D. Song, Curriculum adversarial training, in *Proc. of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 2018, pp. 3740–3747.

[23] I. A. G. Ororbia, C. L. Giles, and D. Kifer, Unifying adversarial training algorithms with flexible deep data gradient regularization, arXiv preprint ar Xiv: 1601.07213, 2016.

[24] C. Song, H. Cheng, H. Yang, S. Li, C. Wu, Q. Wu, Y. Chen, and H. Li, MAT: A multi-strength adversarial training method to mitigate adversarial attacks, in *Proc. of 2018 IEEE Computer Society Annual Symposium on VLSI*, Hong Kong, China, 2018, pp. 476–481.

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

[26] Q. Wang, W. Guo, K. Zhang, I. Ororbia, G. Alexander, X. Xing, X. Liu, and C. L. Giles, Learning adversary-resistant deep neural networks, arXiv preprint arXiv: 1612.01401, 2016.

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

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

[29] J. Wang, J. Sun, P. Zhang, and X. Wang, Detecting adversarial samples for deep neural networks through mutation testing, arXiv preprint arXiv: 1805.05010, 2018.

[30] D. Meng and H. Chen, MagNet: A two-pronged defense against adversarial examples, in *Proc. of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, Dallas, TX, USA, 2017, pp. 135–147.

[31] P. Samangouei, M. Kabkab, and R. Chellappa, Defense-GAN: Protecting classifiers against adversarial attacks using generative models, in *Proc. of 6th International Conference on Learning Representations*, Vancouver, Canada, 2018, p. 714.

[32] W. Xu, D. Evans, and Y. Qi, Feature squeezing: Detecting adversarial examples in deep neural networks, arXiv preprint arXiv: 1704.01155, 2017.

[33] W. Zhou, B. A. Conrad, S. H. Rahim, and E. P. Simoncelli, Image quality assessment: From error visibility to structural similarity, *IEEE Transactions on Image Process*, vol. 13, no. 4, pp. 600–612, 2004.

[34] G. Hinton, O. Vinyals, and J. Dean, Distilling the Knowledge in a neural network, arXiv preprint arXiv: 1503.02531, 2015.

[35] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, Loss functions for image restoration with neural networks, *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, 2017.

[36] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[37] H. Xiao, K. Rasul, and R. Vollgraf, Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms, arXiv preprint arXiv: 1708.07747, 2017.

[38] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, Detection of traffic signs in real-world images: The German traffic sign detection benchmark, in *Proc. of International Joint Conference on Neural Networks*, Dallas, TX, USA, 2013, pp. 1–8.

[39] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv: 1412.6980, 2014.

[40] A. M. Nguyen, J. Yosinski, and J. Clune, Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, 2015, pp. 427–436.

[41] A. Shafahi, W. R. Huang, C. Studer, S. Feizi, and T. Goldstein, Are adversarial examples inevitable? in *Proc. of 7th International Conference on Learning Representations*, New Orleans, LA, USA, 2019, p. 150.

[42] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, Robustness may be at odds with accuracy, in *Proc. of 7th International Conference on Learning Representations*, New Orleans, LA, USA, 2019, p. 1223.

[43] G. F. Elsayed, S. Shankar, B. Cheung, N. Papernot, A. Kurakin, I. J. Goodfellow, and J. Sohl-Dickstein, Adversarial examples that fool both computer vision and time-limited humans, in *Proc. of Advances in Neural Information Processing Systems 31*: *Annual Conference on Neural Information Processing Systems*, Montréal, Canada, 2018, pp. 3914–3924.

Keji Han received the BS degree from Henan University in 2015. He is now a successive master-PhD student at Nanjing University of Posts and Telecommunications. He has published papers in knowledge-based system and pattern recognition. He is a reviewer of *Applied Intelligence*. His research mainly focuses on adversarial machine learning.

Yun Li received the PhD degree in computer science from Chongqing University, Chongqing, China in 2005. He is a professor in the School of Computer Science, Nanjing University of Posts and Telecommunications, China. Prior to that, he was a postdoctoral fellow at the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. He is the Principal Investigator (PI) of several national scientific research projects and provincial projects in recent years. His research mainly focuses on machine learning, data mining, and parallel computing. He has published more than 60 refereed research papers. He is a member of the IEEE.

Bin Xia received the PhD degree from the School of Computer Science and Engineering, Nanjing University of Science and Technology, China in 2018. He is currently an assistant professor at the School of Computer Science, Nanjing University of Posts and Telecommunications. He has published more than 20 papers. His research area includes deep learning, recommender system, and AI for IT operations.