DAFAR: Detecting Adversaries by Feedback-Autoencoder Reconstruction

Haowen Liu and Ping Yi
School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China
{issacliewx, yiping}@sjtu.edu.cn

Hsiao-Ying Lin and Jie Shi
Huawei International, Singapore
{Lin.hsiao.ying, Shi.jie1}@huawei.com

Abstract

Deep learning has shown impressive performance on challenging perceptual tasks. However, researchers found deep neural networks vulnerable to adversarial examples. Since then, many methods are proposed to defend against or detect adversarial examples, but they are either attack-dependent or shown to be ineffective with new attacks.

We propose DAFAR, a feedback framework that allows deep learning models to detect adversarial examples in high accuracy and universality. DAFAR has a relatively simple structure, which contains a target network, a plug-in feedback network and an autoencoder-based detector. The key idea is to capture the high-level features extracted by the target network, and then reconstruct the input using the feedback network. These two parts constitute a feedback autoencoder. It transforms the imperceptible-perturbation attack on the target network directly into obvious reconstruction-error attack on the feedback autoencoder. Finally the detector gives an anomaly score and determines whether the input is adversarial according to the reconstruction errors. Experiments are conducted on MNIST and CIFAR-10 data-sets. Experimental results show that DAFAR is effective against popular and arguably most advanced attacks without losing performance on legitimate samples, with high accuracy and universality across attack methods and parameters.

1. Introduction

Recent years deep learning systems play an increasingly important role in people's everyday life. However, researchers found deep neural networks to be vulnerable to adversarial examples, by applying specially crafted perturbations that are imperceptible to humans on the original samples [7, 9, 23, 29, 34, 43]. The adversarial examples can cause deep learning models to give a wrong classification, which cast a great threat on modern deep learning systems (e.g., automated surveillance cameras [35] and speech recognition systems [37]).

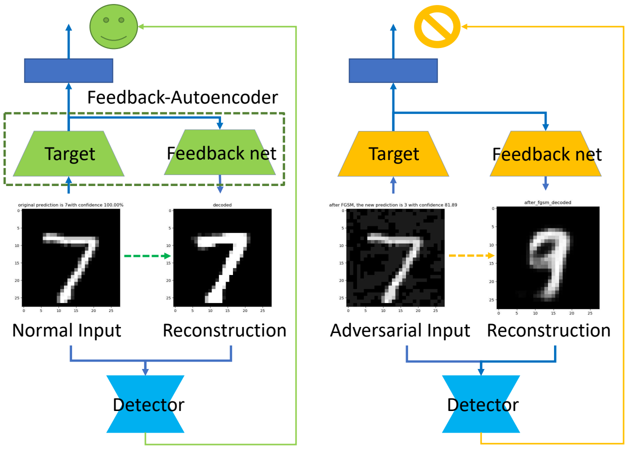

Figure 1. Framework and workflow of DAFAR. Feedback autoencoder reconstructs the input from high-level features, and then the detector gives an anomaly score for the input according to the reconstruction errors and tells whether the input is adversarial. The left is a normal input case while the right shows an adversarial case.

With advanced adversarial attack methods continuing appearing, researches of defense against adversarial examples also have lots of breakthroughs [5, 11, 16, 20, 25, 38]. However, most defense methods either target specific attacks or were shown to be ineffective with new attacks. Some defense methods only focus on properties of specific attack but ignore common properties of adversarial examples (e.g., adversarial training [3] and adversary detector [10, 26]), leading to attack-dependent. Other defense methods with relatively high universality are either easily broken down or with complex mechanism (e.g., Gradient Masking [11], Input Transformation [5] and MagNet [25]).

We propose DAFAR[1], shown in Figure 1, a framework to detect adversarial examples with three significant advantages. First, the architecture is simple. It only contains a feedback network and an autoencoder-based detector besides the target network, and the feedback network often can be transformed from the byproduct of target network's pre training due to its autoencoder-basis [14, 17, 24]. Second, it does not modify the target network, for the feedback network is a plug-in structure, and the detector is an independent part, so it can be used to protect a wide range of neural networks. Third, DAFAR's target network and feedback network are trained on normal samples and its detector is trained on reconstruction errors of normal samples in semi supervised way, so DAFAR is attack-independent.

Our main contributions are:

- We summarize two principles to achieve ideal adversary detection effectiveness (Section 3).
- Based on the two principles, we design DAFAR, a semisupervised-trained framework to detect adversarial examples with high accuracy and universality (Section 4, 5).
- We combine DAFAR with Image Filter to achieve ideal defense effectiveness (Section 6.1). Based on our work, we advocate that a combination of only-detection method and complete-defense method is promising to research on to achieve ideal defense effectiveness against adversarial examples (Section 6.2).

2. Background

2.1. Deep Learning

Deep learning is a type of machine learning methods based on deep neural networks (DNNs) [21], which makes information systems to learn knowledge without explicit programing and extract useful features from raw data. Deep learning models play an increasingly important role in modern life. They are used in image classification [22, 30], financial analysis [27], disease diagnosis [6], speech recognition [32] and information security [2, 39].

Several popular network structures are used in daily life and research widely: LeNet [19], AlexNet [18], VGGNet [31], GoogLeNet [33] and ResNet [12]. Attackers usually generate adversarial examples against these baseline architectures [40].

2.2. Adversarial Examples

Adversarial examples are original clean samples with specially crafted small perturbations, often barely recognizable by humans, but able to misguide the classifier. Since the discovery of adversarial examples for neural networks in [34], researchers have developed several methods to generate adversarial examples, such as fast gradient sign method (FGSM) [9], Carlini and Wagner Attacks (CW) [7], Jacobian-based Saliency Map Attack (JSMA) [29], and Projected Gradient Descent (PGD) [23]. We will use these four popular and arguably most advanced attack methods in our experiments later.

2.3. Typical Defense Methods

Adversarial example defense can be categorized into two types: *Only Detection* methods and *Complete Defense* methods [1]. In this section we briefly summarize some typical defense methods in these two categories separately.

Only Detection. *Only-binary-classifier method* [26] simply trains a supervised binary classifier on normal and adversarial samples to classify them. *MagNet* [25] contains an autoencoder-based detector to detect adversaries with large reconstruction errors and a probability-divergence-based detector to handle small ones. *Feature Squeezing* [38] detects adversarial examples by comparing a DNN model's prediction on the original input with that on squeezed inputs. These methods are either attack-dependent or easily cheated by low-intensity attacks, where DAFAR shows a better performance.

Complete Defense. *Adversarial training* techniques [3] train a more robust model by including adversarial information in training process. *Gradient Masking* techniques [11] reduce the sensitivity of DNN models to small changes in inputs to defend against gradient-based attacks. *Input Transformation* techniques [5] like Image Filter reduce the model sensitivity to small input changes by transforming the inputs, relieving or changing the input changes. These methods either lose effectiveness in some cases or are easily broken by high-intensity attacks, which is exactly what DAFAR is good at.

3. Motivation and Goal

Despite the considerable research effort expended towards defending against adversarial examples, scientific literature still lacks universal and effective methods to defend against adversarial examples.

To achieve ideal effectiveness for an adversary detection, MagNet and Feature Squeezing provide preliminary examples. We advocate two principles to achieve ideal detection effectiveness.

1. Instead of focusing on properties of adversarial examples from specific generation processes, find intrinsic common properties among all adversarial examples across attack methods and parameters. This principle will lead to attack-independent detection and defense.

[1] DAFAR is the abbreviation for **D**etecting **A**dversaries by **F**eedback-**A**utoencoder **R**econstruction.

2. Amplify the difference between normal sample and adversarial example as much as possible, and detect adversarial examples according to that difference. This principle will lead to high detection accuracy.

Our goal is to design an adversary detection framework based on the above two principles, which can achieve high accuracy as well as universality across attack methods and parameters. To do so, we propose DAFAR according to common properties of adversarial examples. We carry out empirical experiments to validate and demonstrate the rationale of our ideas. We also conduct experiments to evaluate DAFAR by comparing it with several typical adversary detection methods. Finally we propose a promising direction to achieve ideal defense effectiveness against adversarial examples according to our work.

4. Design

4.1. High-level Feature Interference

Whether gradient-based or optimization-based attacks misguide the classifier by adding specially crafted small perturbations to clean samples. Deep learning models determine the label of a sample by extract the high-level features of the sample at deep layers [21]. Adversarial perturbations significantly interfere in the feature extraction process, inducing huge disturbance into high-level features extracted, leading to unexpected change of feature semantics, ultimately causing the classifier to mis-classify. Formally, given $f(\cdot)$ as the target classifier, $E(\cdot)$ as feature extraction layers of f, $F(\cdot)$ as the output layers of f, x as a normal sample, and x' as an adversarial sample of x, the process that x' misguides f can be described as

$$\begin{aligned} \delta(E(x), E(x')) \gg \delta(x, x') \\ F(E(x')) \neq F(E(x)) \end{aligned} \tag{1}$$

where $\delta(a, b)$ means the difference between a and b for a given distance function $\delta(\cdot, \cdot)$. We will demonstrate the high-level feature interference caused by adversary in Section 5.2.1 with experiments.

4.2. Reconstruction Errors

In Formula 1 we find an inequality, $\delta(E(x), E(x')) \gg \delta(x, x')$, corresponding to *Principle 2* that to detect adversarial examples with high accuracy the amplification of the difference between normal sample and adversarial example is necessary. However, only using this inequality can do nothing because we only have a normal sample or an adversarial example when inputing a sample into a model.

What we have when detecting is an input sample x, deep neural network f and high-level features $E(x)$. According to [4, 14, 36], an autoencoder reconstructs a sample using the features extracted by its encoder (feature extraction layers). If the features extracted by encoder are disturbed, the decoded output will present a significant *reconstruction error*. This significant reconstruction error, or called *reconstruction distance* between reconstruction sample and original sample is exactly what we want, according to *Principle 2*. So we can add a feedback decoder $D(\cdot)$ to the feature extraction layers $E(\cdot)$ of target network to reconstruct the high-level features $E(x)$ to a reconstruction sample $D(E(x))$. The target network $E(\cdot)$ and the feedback network $D(\cdot)$ constitute a feedback autoencoder $D(E(\cdot))$. This structure transforms the attack on target network (imperceptible perturbations) into an obvious attack on the feedback autoencoder (reconstruction error) directly. After appropriate training, the normal samples will be reconstructed perfectly with small reconstruction errors, while the adversarial ones which attack the target network will present significant reconstruction errors because they *attack the autoencoder as well*. Formally we can give a description by

$$\delta(x', D\left(E(x')\right)) \gg \delta(x, D\left(E(x)\right)) \tag{2}$$

where $\delta(x, D\left(E(x)\right))$ is the reconstruction distance of sample x, which can be described as

$$\delta(x, D\left(E(x)\right) = ||x - D\left(E(x)\right)||_p \tag{3}$$

In this way we greatly amplify the difference between a normal sample and an adversarial example. By detecting the difference, we can detect adversarial examples with higher accuracy. Moreover, since the amplification is based on the common properties among all adversarial examples, according to *Principle 1*, this method is attack-independent. In other words, DAFAR is able to detect adversarial examples with high accuracy and universality. More formally we describe detection process of DAFAR by

$$O(x) = F(E(x)) \wedge C\left(\delta(x, D\left(E(x)\right))\right) \tag{4}$$

where $C(\cdot)$ is the detector that judges a sample whether legitimate or adversarial by the reconstruction errors.

Compared to the first detector of MagNet, DAFAR detects the disturbance caused by adversarial examples in the target network as much as possible, for it directly changes the attack on target network to the attack on the feedback autoencoder, avoiding missing low-intensity adversary due to robustness of the neural network itself.

4.3. DAFAR Structure

We discuss the detailed structure and workflow of DAFAR in this section, which is shown in Figure 2.

4.3.1 Target Network

A target network is the deep learning model facing adversarial attacks directly, shown as $F(E(\cdot))$ in Figure 2.

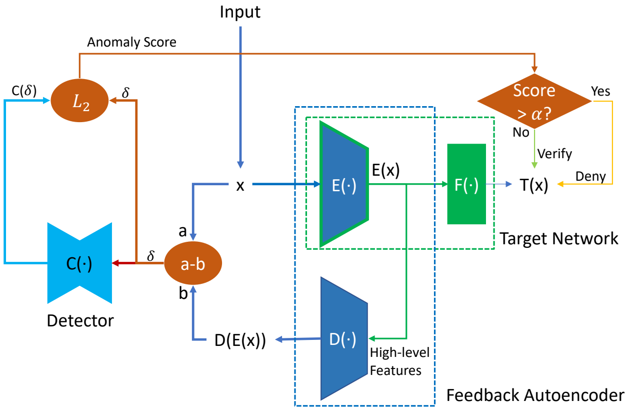

Figure 2. Detailed structure and workflow of DAFAR. Feedback network $D(\cdot)$ reconstructs the input x from high-level features $E(x)$, and then the detector $C(\cdot)$ gives an anomaly score in the form of L_2 distance for the input x, according to the reconstruction errors δ, and judges whether the input is adversarial by comparing the score with a threshold α.

The encoder $E(\cdot)$ of feedback autoencoder $D(E(\cdot))$ is exactly the feature extraction layers $E(\cdot)$ of the target network $F(E(\cdot))$. This part of DAFAR transforms the disturbance in the feature extraction layers of target network directly into disturbance in the encoder of feedback autoencoder, which will cause significant reconstruction errors in the decoder $D(\cdot)$ later.

4.3.2 Decoder/Feedback Network

The feedback network, which is also called decoder and shown as $D(\cdot)$ in Figure 2, reconstructs the input x from high-level features extracted by the encoder ($E(x)$). Though the target network and the decoder are two separated parts, to ensure the reconstruction quality and the classifying accuracy of target classifier, we train target network and feedback network at the same time by minimizing a loss function over the training set, where the loss function is the combination of cross entropy function loss and mean squared error loss

$$J_f(\mathbb{X}_{train}) = \frac{1}{\#\mathbb{X}_{train}} \cdot \sum_{x \in \mathbb{X}_{train}} \left(||x - D(E(x))||_2 + \sum p(x) log \frac{1}{F(E(x))} \right) \tag{5}$$

where $\mathbb{X}_{train}$ is the training dataset only containing normal samples, and $p(x)$ is the grand truth label vector of sample x. Training in this manner will not affect the accuracy of original target network, for autoencoder is often used in pre training and shows effective in learning representations for subsequent classification tasks [17, 24], which we will demonstrate in our experiments later.

4.3.3 Detector

As we mentioned above, DAFAR distinguishes adversarial examples by detecting significant reconstruction distance. We use an anomaly detection autoencoder as the decoder, shown as $C(\cdot)$ in Figure 2. Autoencoder is often used in anomaly detection [41,42,44]. In most cases, the number of adversarial examples is much less than normal samples, so adversarial examples are an *anomaly*. However, since the difference between normal samples and adversarial examples is very small, directly applying anomaly detection autoencoder to adversary detection is not recommended. Yet in our case, we have greatly amplified the difference between adversarial examples and normal samples by feedback autoencoder, in the form of reconstruction errors. So it is reasonable to use anomaly detection autoencoder as our final detector. To this end, we use *image subtraction* ($a - b$ in Figure 2) as reconstruction errors. Given $\mathbb{E}_{train}$ as the the training set only containing reconstruction errors of normal samples, and δ as the reconstruction errors of a normal sample in $\mathbb{E}_{train}$, we train the detector $C(\cdot)$ by minimizing mean squared error loss in a semi supervised learning manner

$$J_C\left(\mathbb{E}_{train}\right) = \frac{1}{\#\mathbb{E}_{train}} \cdot \sum_{\delta \in \mathbb{E}_{train}} ||\delta - C(\delta)||_2 \tag{6}$$

After training, the detector $C(\cdot)$ will reconstruct normal data well, while failing to do so for anomaly data which the detector $C(\cdot)$ has not encountered. The detector $C(\cdot)$ does not tell us whether a reconstruction error belongs to an adversarial example or not directly. It only gives us a reconstruction distance between its input and output, which is used as ***anomaly score***. Input with high anomaly score is considered to be an anomaly. In order to define whether an anomaly score is high or low, a threshold α is needed. We set the 99.7% confidence interval's right edge calculated from anomaly scores of all normal samples in training set as α for MNIST and 95% for CIFAR-10. Given $\bar{x}$ as the average score, σ as the standard deviation and n as the number of normal samples in training set, then α is

$$\alpha = \bar{x} + z \cdot \frac{\sigma}{n}, \quad z = 2, 3 \tag{7}$$

For other datasets, people can set the threshold with an appropriate confidence interval according to real needs, in the same manner as MNIST and CIFAR-10 here.

5. Experiments

5.1. Network Structure

In this section we describe network structures used in our experiments.

Target network. We choose practical network structures as target networks, which face the adversarial attacks directly. Though we have mentioned the encoder is the feature extraction layers of target network, actually the encoder does not have to include all feature extraction layers. It can be just several former layers, according to real needs. In other words, we can determine the layers to capture high-level features (i.e., feedback positions) according to what trade-off we want to make between training overhead and detection effectiveness.

MNIST $\mathcal{T}_M$		CIFAR-10 $\mathcal{T}_C$	
Encoder $\mathcal{E}_M$		Encoder $\mathcal{E}_C$	
Conv.ReLU	$3 \times 3 \times 32$	Conv.ReLU	$3 \times 3 \times 96$
Conv.ReLU	$3 \times 3 \times 32$	Conv.ReLU	$3 \times 3 \times 96$
MaxPool	2×2	Conv.ReLU	$3 \times 3 \times 96$
Conv.ReLU	$3 \times 3 \times 64$	MaxPool	2×2
Conv.ReLU	$3 \times 3 \times 64$	Conv.ReLU	$3 \times 3 \times 192$
MaxPool	2×2	Conv.ReLU	$3 \times 3 \times 192$
		Conv.ReLU	$3 \times 3 \times 192$
		MaxPool	2×2
Output $\mathcal{F}_M$		Output $\mathcal{F}_C$	
Linear.ReLU	200	Conv.ReLU	$3 \times 3 \times 192$
Linear.ReLU	200	Conv.ReLU	$1 \times 1 \times 192$
Softmax	10	Conv.ReLU	$1 \times 1 \times 10$
		Linear.ReLU	200
		Linear.ReLU	200
		Softmax	10

Table 1. Structures of target networks.

Decoder/Feedback network. The structures of decoders are often the reverse of their encoders. But they can be modified according to real needs, such as transforming from the byproduct network in unsupervised pre training.

Detector. Detectors are simple 7-layer fully connected autoencoders for MNIST and CIFAR-10. We do not elaborately choose structures here, but experimental results later is outstanding, showing the effectiveness of DAFAR.

As we have mentioned above, the DAFAR structure is a relatively simple structure and easy to implement or remove. The complexity of DAFAR is proportional to the complexity of target network.

5.2. Reconstruction Error: Why Does DAFAR Work?

In this section we conduct experiments to characterize the phenomenon of reconstruction errors in DAFAR structure, to explain why DAFAR works.

5.2.1 High-level Feature Interference

As we discussed in Section 4.1, whether gradient-based or optimization-based attacks induce huge disturbance into high-level features, leading to unexpected changes of feature semantics. In order to show high-level feature interference caused by adversarial perturbations, we carry out experiments by extracting high-level features of adversarial examples and their original normal samples from deep layers of target network and calculating distance between them. We compare the result with the distance between high-level features of samples added same-intensity Gaussian noise and that of their original samples in Table 2.

Perturbations	Gaussian $(0.3, 0.3)$	FGSM 0.3	PGD 0.3
L_2 distance	63.25	102.64	96.44

Table 2. Average high-level features L_2 distance between samples with certain perturbations and their original samples, of 1000 samples in MNIST test set.

Clearly adversarial perturbations cause a much bigger interference to high-level features of a sample. Here we can draw an observation.

Observation 1. *Adversarial perturbations induce huge disturbance into high-level features extracted by deep layers of target network.*

5.2.2 Reconstruction Errors

The interference in adversarial examples' high-level features will lead to big reconstruction errors, as we discussed in Section 4.2. In this section we will give more details on characterizations of reconstruction errors.

Figure 3 shows the significant difference between the reconstruction errors of normal samples and that of adversar-

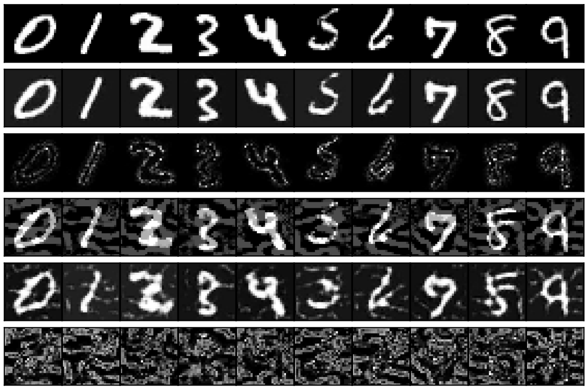

Figure 3. Reconstruction errors of MNIST normal and adversarial samples. The first two lines are separately normal samples and their reconstructions, and the third line are their reconstruction errors. The forth and fifth lines are separately adversarial examples and their reconstructions, and the last line are their reconstruction errors. There is significant difference between the third and last lines, in the aspects of errors size and patterns.

ial examples clearly. To quantitatively characterize how reconstruction errors distribute with attack methods and intensities, we calculate the reconstruction distance of normal samples and adversarial examples across different attack methods and intensities in the form of L_2 distance. The results are shown in Figure 4.

Here we can draw three observations.

Observation 2. *Difference of reconstruction errors between normal and adversarial samples is significant in two aspects: 1) reconstruction errors of normal and adversarial samples show very different patterns; 2) reconstruction errors in the form of L_2 shows distinctively separated concentrations across attack intensities.*

Observation 3. *Though the difference is clear, there is no fixing reconstruction-error threshold to perfectly divide normal samples and adversarial examples, especially at low attack intensity.*

Observation 4. *Difference of reconstruction errors between adversarial examples and normal samples increases with attack intensities.*

5.2.3 Anomaly Score

As we discussed in Section 4.3.3, detector needs a threshold to tell whether an input is an anomaly, so there should be a clear dividing line of anomaly score between normal inputs and anomalies. We train a detector in semi supervised manner on clean samples' reconstruction errors. Then we input clean samples and adversarial examples across different attack methods and intensities to calculate their anomaly scores. Figure 5 shows how anomaly scores distribute with attack methods and intensities. Here we draw two important observations.

Observation 5. *The distribution of anomaly scores is approximately normal, so we assume that the score follows a normal distribution. We set the* 99.7% *confidence interval's right edge of normal samples' anomaly scores for MNIST and* 95% *for CIFAR-10 as the threshold to distinguish normal and adversarial samples, as discussed in Section 4.3.3. We show the score threshold of MNIST and CIFAR-10 in Table 3, and also show in Figure 5.*

Dataset	Score threshold
MNIST	23.333
CIFAR-10	230.143

Table 3. Score threshold of MNIST and CIFAR-10.

Observation 6. *Detector further magnifies the difference between normal and adversarial samples. There is a clear dividing line of anomaly score between normal input and anomaly, which means it is reasonable to determine a threshold to tell whether an input is an anomaly.*

We hypothesize that detector's secondary amplification effect is introduced by two reasons: 1) reconstruction distances of adversarial examples with large intensities are much bigger than that of normal samples, which detector can easily distinguish; 2) even if the reconstruction-distance difference between normal and adversarial samples is not that much, their reconstruction errors show very different patterns as shown in Figure 3, which detector can refer to.

5.3. Evaluation

In this section we evaluate the accuracy and universality of DAFAR in detecting adversarial examples using FGSM, JSMA, CW_2 and PGD across different attack intensities, and compare the results with only-binary-classifier method, detection system of MagNet and Feature Squeezing. For FGSM, PGD and CW_2, we used the implementation of Cleverhans [28]. For JSMA, we use authors' open source implementation [29]. And we also use authors' open source implementations to implement detector system of MagNet and Feature Squeezing. [25, 38].

In principle, DAFAR shows a better performance of accuracy than MagNet and Feature Squeezing, especially in low attack intensities, and the same level of universality as MagNet and Feature Squeezing across attack methods, which is much better than only-binary-classifier method.

5.3.1 Accuracy and Universality across Attack Intensities

In this section we evaluate DAFAR's adversary-detection accuracy and universality across different attack intensities, separately on MNIST and CIFAR-10, using FGSM attack across different attack intensities.

MNIST. We train a target network in DAFAR method on MNIST and achieve an accuracy of 99.24% on the test set, which is close to the state of the art. We test the adversary detection accuracy of each method on test sets only containing FGSM adversarial examples across different attack intensities. The results are shown in Figure 6. Here we can draw some conclusions.

Effect on normal examples. The target network trained in DAFAR method achieves an accuracy of 99.24%, and the detector of DAFAR shows a false positive rate of only 0.16%, which means DAFAR does not affect target network's accuracy.

Effect on adversarial examples. DAFAR detects MNIST adversarial examples in an accuracy of 100% across all attack intensities, as shown in Figure 6, higher than other three methods especially in low attack intensities, showing the best detection accuracy and universality.

CIFAR-10. CIFAR-10 is a much more complex dataset

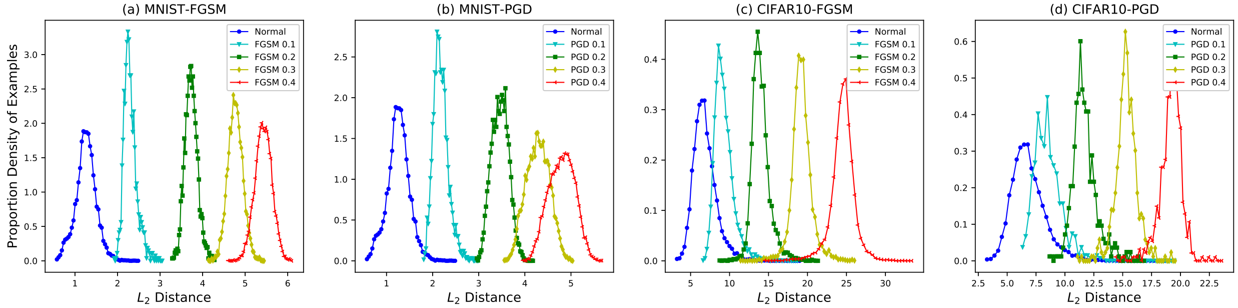

Figure 4. Reconstruction-distance proportion density curves of adversarial examples and normal samples, across different attack methods and intensities. Every peak shows the area of L_2 distance concentration of a certain attack intensity. The separated peaks show reconstruction distance increases with attack intensity, which also causes adversarial examples differentiated from normal samples. However, in CIFAR-10 the distinction between the normal samples' peak and low-intensity adversarial examples' peak is not that clear.

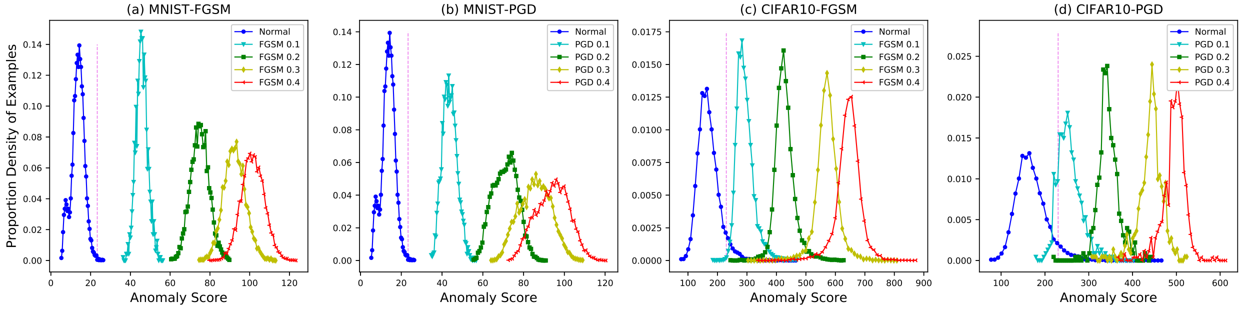

Figure 5. Anomaly-score proportion density curves of adversarial examples and normal samples, across different attack methods and intensities. Every peak shows the area of anomaly score concentration of a certain attack intensity. Violet vertical line indicates the location of the anomaly score threshold, which clearly divides normal samples and adversarial examples.

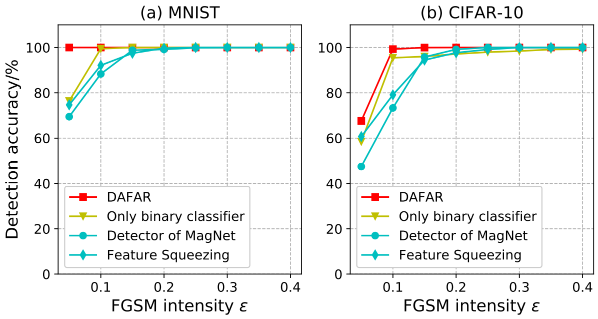

Figure 6. Detection accuracy curves on MNIST and CIFAR-10 adversarial examples across different FGSM intensities.

than MNIST. We train a target network in DAFAR method on CIFAR-10 and achieve an accuracy of 86.17% on the test set, which is at the normal level. We test the adversary detection accuracy of each method on test sets only containing FGSM adversarial examples across different attack intensities. The results are shown in Figure 6. Here we can draw some conclusions.

Effect on normal examples. The target network trained in DAFAR method on CIFAR-10 achieves an accuracy of 86.17%, and the detector of DAFAR shows a false positive rate of only 3.49% on normal samples. It is a negligible performance reduction.

Effect on adversarial examples. DAFAR detects CIFAR-10 adversarial examples in an accuracy of 100% across most of attack intensities, as shown in Figure 6, but not as accurate as on MNIST when the attack intensity is very low. However, it is still higher than other three methods especially in low attack intensities. This also provides empirical evidence that DAFAR achieves the best detection effectiveness across different attack intensities.

5.3.2 Accuracy and Universality across Attack Methods

In this section we evaluate DAFAR's detection accuracy and universality across different attack methods, separately on MNIST and CIFAR-10, using FGSM, JSMA, CW_2 and PGD.

MNIST. We test the adversary detection accuracy of each method on MNIST test sets containing adversarial examples across different attack methods. Figure 7 shows the results, which provides evidence that DAFAR has the same level of universality across different attack methods as detection system of MagNet and Feature Squeezing, much better than only-binary-classifier method.

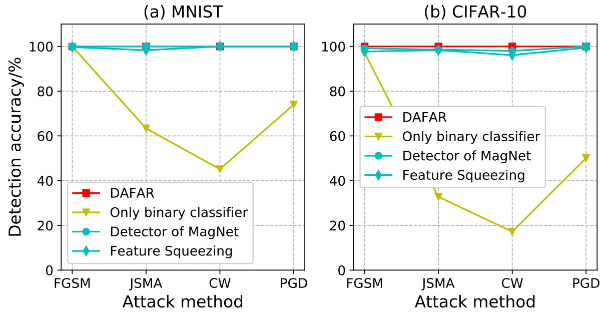

Figure 7. Detection accuracy curves on MNIST and CIFAR-10 adversarial examples across different attack methods.

CIFAR-10. We test the adversary detection accuracy of each method on CIFAR-10 test sets containing adversarial examples across different attack methods. The results are shown in Figure 7, which give the same conclusion as MNIST's.

For all parts in DAFAR are trained in semi supervised way or only on clean samples, theoretically DAFAR has outstanding universality across attack methods, which is strongly proved by our experimental results. Note that we achieve that outstanding performance without using complex autoencoder structures.

6. Adversary Defense Going Forward

6.1. DAFAR Combined with Image Filter

Though DAFAR shows a relatively good performance, it still does not achieve ideal detection accuracy at very low attack intensities. There are two ways to approach ideal defense effectiveness. First, train DAFAR in more appropriate parameters, network architectures and training methods, to compress anomaly score interval of normal samples (i.e., the blue peaks in Figure 5) as much as possible until the interval converges to 0, which is the ideal condition, but very difficult to achieve. Second, instead of optimizing DAFAR, choose others to help with the problem DAFAR is not good at.

Fortunately there are a few methods very effective for low attack intensity, such as image filter [13], denoising autoencoder [8, 15] and adversarial training [3]. They are easily broken down by high-intensity attacks, but good helpers for DAFAR at low attack intensities. For example, we combine DAFAR with image filter. First DAFAR discards the inputs judged adversarial, then image filter filters the inputs that pass DAFAR's judgement, and inputs the filtered samples into target classifier. We carry out an experiment on CIFAR-10 datasets in which the ratio of adversarial examples and normal samples is $1:1$ to evaluate this structure, and show the results in Figure 8. The classifying accuracies across all attack intensities are near 86%, very close to the original accuracy of target classifier, presenting ideal defense effectiveness.

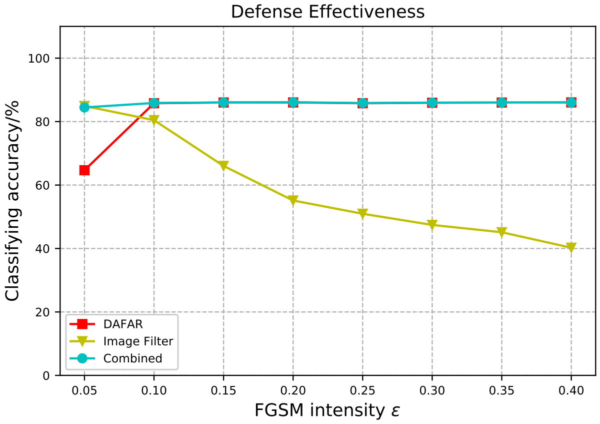

Figure 8. Classifying accuracy curves on CIFAR-10 adversarial and normal samples $(1:1)$.

6.2. How to Achieve Ideal Defense Effectiveness?

The previous section provides an example that combining *Only Detection* method and *Complete Defense* method can achieve ideal defense effectiveness. Actually it is not an individual case.

Only-detection is good at detecting high-intensity attacks but can be cheated by low-intensity attacks, while complete-defense is effective for low-intensity attacks but easily broken down by high-intensity attacks [13]. So we believe that first using only-detection to discard high-intensity attacks and then applying complete-defense to eliminate low-intensity attacks can lead to ideal defense effectiveness. A strong detector with a weak defender like DAFAR combined with image filter, or the reverse, is worth researching on.

7. Conclusion

DAFAR is a feedback framework that helps deep learning models to detect adversarial examples effectively. Besides the target network, DAFAR only contains a plug-in feedback network and an autoencoder-based decoder. The former amplifies the perturbations of adversarial examples in the form of reconstruction errors, and the latter secondarily amplifies the reconstruction errors, making perturbations easily detected. Considering all parts in DAFAR are trained on normal samples or in semi supervised way, DAFAR is attack-independent. Our experiments explain DA-

FAR's work mechanism and show DAFAR can detect state-of-art attacks with high accuracy and universality.

According to our work, we advocate that a combination of only-detection method and complete-defense method is promising to research on to achieve ideal defense effectiveness against adversarial examples.

References

[1] N. Akhtar and A. Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430, 2018. 2

[2] Abdulrahman Al-Abassi, Hadis Karimipour, Ali Dehghantanha, and Reza M Parizi. An ensemble deep learning-based cyber-attack detection in industrial control system. *IEEE Access*, 8:83965–83973, 2020. 2

[3] Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. *Advances in Neural Information Processing Systems*, 33, 2020. 1, 2, 8

[4] Marilyn Bello, Gonzalo Nápoles, Ricardo Sánchez, Rafael Bello, and Koen Vanhoof. Deep neural network to extract high-level features and labels in multi-label classification problems. *Neurocomputing*, 413:259 – 270, 2020. 3

[5] Arjun Nitin Bhagoji, Daniel Cullina, Chawin Sitawarin, and Prateek Mittal. Enhancing robustness of machine learning systems via data transformations. In *2018 52nd Annual Conference on Information Sciences and Systems (CISS)*, pages 1–5. IEEE, 2018. 1, 2

[6] Xiuli Bi, Shutong Li, Bin Xiao, Yu Li, Guoyin Wang, and Xu Ma. Computer aided alzheimer's disease diagnosis by an unsupervised deep learning technology. *Neurocomputing*, 392:296–304, 2020. 2

[7] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017. 1, 2

[8] S. Cho, T. J. Jun, B. Oh, and D. Kim. Dapas : Denoising autoencoder to prevent adversarial attack in semantic segmentation. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020. 8

[9] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *Computer ence*, 2014. 1, 2

[10] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017. 1

[11] Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014. 1, 2

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 2

[13] Warren He, James Wei, Xinyun Chen, Nicholas Carlini, and Dawn Song. Adversarial example defense: Ensembles of weak defenses are not strong. In *11th USENIX Workshop on Offensive Technologies (WOOT 17)*, Vancouver, BC, Aug. 2017. USENIX Association. 8

[14] Geoffrey E Hinton. Learning multiple layers of representation. *Trends in cognitive sciences*, 11(10):428–434, 2007. 2, 3

[15] U. Hwang, J. Park, H. Jang, S. Yoon, and N. I. Cho. Puvae: A variational autoencoder to purify adversarial examples. *IEEE Access*, 7:126582–126593, 2019. 8

[16] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, and Hassan Foroosh. Comdefend: An efficient image compression model to defend adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[17] Silvija Kokalj-Filipovic, Rob Miller, Nicholas Chang, and Chi Leung Lau. Mitigation of adversarial examples in rf deep classifiers utilizing autoencoder pre-training. In *2019 International Conference on Military Communications and Information Systems (ICMCIS)*, pages 1–6. IEEE, 2019. 2, 4

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 2

[19] Quoc V Le. Building high-level features using large scale unsupervised learning. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 8595–8598. IEEE, 2013. 2

[20] Wenqing Liu, Miaojing Shi, Teddy Furon, and Li Li. Defending adversarial examples via dnn bottleneck reinforcement. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1930–1938, 2020. 1

[21] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E. Alsaadi. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11 – 26, 2017. 2, 3

[22] Benteng Ma, Xiang Li, Yong Xia, and Yanning Zhang. Autonomous deep learning: A genetic dcnn designer for image classification. *Neurocomputing*, 379:152–161, 2020. 2

[23] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 1, 2

[24] Huzaifa M. Maniyar, Nahid Guard, and Suneeta V. Budihal. Stacked denoising autoencoder: A learning-based algorithm for the reconstruction of handwritten digits. In Chhabi Rani Panigrahi, Bibudhendu Pati, Prasant Mohapatra, Rajkumar Buyya, and Kuan-Ching Li, editors, *Progress in Advanced Computing and Intelligent Engineering*, pages 377–387, Singapore, 2021. Springer Singapore. 2, 4

[25] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 135–147, 2017. 1, 2, 6

[26] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*, 2017. 1, 2

[27] Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer. Deep learning for financial applications: A survey. *Applied Soft Computing*, page 106384, 2020. 2

[28] Nicolas Papernot, Ian Goodfellow, Ryan Sheatsley, Reuben Feinman, and Patrick Mcdaniel. cleverhans v1.0.0: an adversarial machine learning library. 2016. 6

[29] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016. 1, 2, 6

[30] K Shankar, Yizhuo Zhang, Yiwei Liu, Ling Wu, and Chi-Hua Chen. Hyperparameter tuning deep learning for diabetic retinopathy fundus image classification. *IEEE Access*, 8:118164–118173, 2020. 2

[31] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *Computer ence*, 2014. 2

[32] Zhaojuan Song. English speech recognition based on deep learning with multiple features. *Computing*, 102(3):663–682, 2020. 2

[33] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*, 2016. 2

[34] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *Computer ence*, 2013. 1, 2

[35] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1

[36] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1096–1103, New York, NY, USA, 2008. Association for Computing Machinery. 3

[37] Donghua Wang, Li Dong, Rangding Wang, Diqun Yan, and Jie Wang. Targeted speech adversarial example generation with generative adversarial network. *IEEE Access*, 8:124503–124513, 2020. 1

[38] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *Network and Distributed System Security Symposium*, 2017. 1, 2, 6

[39] Chao-Tung Yang, Jung-Chun Liu, Endah Kristiani, Ming-Lun Liu, Ilsun You, and Giovanni Pau. Netflow monitoring and cyberattack detection using deep learning with ceph. *IEEE Access*, 8:7842–7850, 2020. 2

[40] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9):2805–2824, 2019. 2

[41] S. Zavrak and M. İskefiyeli. Anomaly-based intrusion detection from network flow features using variational autoencoder. *IEEE Access*, 8:108346–108358, 2020. 4

[42] F. Zhang and H. Fleyeh. Anomaly detection of heat energy usage in district heating substations using lstm based variational autoencoder combined with physical model. In *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 153–158, 2020. 4

[43] Huangzhao Zhang, Hao Zhou, Ning Miao, and Lei Li. Generating fluent adversarial examples for natural languages. *arXiv preprint arXiv:2007.06174*, 2020. 1

[44] Qiang Zhao and Fakhri Karray. Anomaly detection for images using auto-encoder based sparse representation. In Aurélio Campilho, Fakhri Karray, and Zhou Wang, editors, *Image Analysis and Recognition*, pages 144–153, Cham, 2020. Springer International Publishing. 4