IEICE TRANSACTIONS

on Information and Systems

VOL. E103-D NO. 4
APRIL 2020

A PUBLICATION OF THE INFORMATION AND SYSTEMS SOCIETY
The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER

Evaluating Deep Learning for Image Classification in Adversarial Environment

Ye PENG[†], ***Student Member***, **Wentao ZHAO**[†], **Wei CAI**[††], **Jinshu SU**[†], **Biao HAN**[†], ***and*** **Qiang LIU**[†a)], ***Nonmembers***

SUMMARY Due to the superior performance, deep learning has been widely applied to various applications, including image classification, bioinformatics, and cybersecurity. Nevertheless, the research investigations on deep learning in the adversarial environment are still on their preliminary stage. The emerging adversarial learning methods, e.g., generative adversarial networks, have introduced two vital questions: to what degree the security of deep learning with the presence of adversarial examples is; how to evaluate the performance of deep learning models in adversarial environment, thus, to raise security advice such that the selected application system based on deep learning is resistant to adversarial examples. To see the answers, we leverage image classification as an example application scenario to propose a framework of Evaluating Deep Learning for Image Classification (EDLIC) to conduct comprehensively quantitative analysis. Moreover, we introduce a set of evaluating metrics to measure the performance of different attacking and defensive techniques. After that, we conduct extensive experiments towards the performance of deep learning for image classification under different adversarial environments to validate the scalability of EDLIC. Finally, we give some advice about the selection of deep learning models for image classification based on these comparative results.

key words: *adversarial environment, deep learning, evaluating metrics, image classification, security evaluation*

1. Introduction

Deep learning and its variants have been emerging as promising tools in many applications such as image classification, bioinformatics, cybersecurity, etc [1]–[4]. However, existing works have revealed that deep learning is prone to a variety of adversarial examples [5]. Taking image classification as an example, the emerging adversarial learning methods, e.g, generative adversarial networks, can effectively generate high-fidelity pictures that are categorized into one group of pictures by human-beings but another by machine learning systems. Even if a black-box adversary has no prior knowledge regarding the details of deep learning models including specific parameters and algorithms, he/she is able to figure out model behaviors by feeding inputs into the models and observing corresponding outputs [6]. As a result, the performance of deep learning based image classification in terms of classification accuracy is significantly reduced. In fact, we encounter two vital questions: (a) to what degree the security of deep learning with the presence of adversarial examples is? and (b) how to evaluate the performance of deep learning models in the adversarial environment, thus, to raise security advice such that the selected application system based on deep learning is resistant to adversarial examples?

Although many works about secure machine learning in the adversarial environment have been proposed [7], the study on security assessment towards deep learning in the adversarial environment is still in its fancy. As indicated in our previous survey [8]–[10], a security assessment is one of the major research points that researchers from both the cybersecurity and the artificial intelligence domains concern. Basically, a security assessment can be investigated from four aspects: assessment methodology, evaluating metrics, alternative adversarial models and available defensive techniques. Specifically, the assessment methodology mainly answers how we evaluate the security of deep learning with the presence of adversarial examples [11]. Due to the fact that there are many alternative threat models and defensive techniques under different assumptions, a good security assessment mechanism should also give a comprehensive guideline regarding how to select a proper adversarial model or defensive method to maximize its effectiveness given a target deep learning model.

To address the above concerns, we select image classification as an application example and propose a framework of Evaluating Deep Learning for Image Classification (EDLIC) to conduct comprehensively quantitative analysis on a variety of threat models and defensive methods of deep learning in an adversarial environment. It is worth noticing that the proposed EDLIC framework is also scalable to other application scenarios given corresponding training and testing examples. Moreover, we introduce a set of evaluating metrics covering capacity and overhead to compare the effectiveness of different attacking and defensive techniques. Then, we conduct extensive comparative experiments over well-known benchmark datasets towards the performance of deep learning under different settings of adversarial environments with alternative combinations of attacking and defensive methods. Based on the above comparative results, we finally give some recommendations about good choices of defensive methods to enhance the robustness of deep learning based image classification systems against adversaries. The main contributions of this paper stem from the follow-

Manuscript received July 4, 2019.
Manuscript revised November 15, 2019.
Manuscript publicized December 23, 2019.

[†]The authors are with the College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China.
[††]The author is with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, Guangdong 518172, China.
a) E-mail: qiangliu06@nudt.edu.cn (Corresponding author)

DOI: 10.1587/transinf.2019EDP7188

ing three aspects:

- We propose the EDLIC framework and apply it to evaluate the performance of deep learning for image classification under different settings of adversarial environments;
- We present a set of evaluating metrics to compare the effectiveness of different attacking and defensive techniques;
- We conduct extensive experiments to validate the effectiveness of the EDLIC framework and find out some valuable suggestions to enhance the robustness of deep learning based image classification systems against existing attacks.

The remainder of this paper is organized as follows: Sect. 2 presents the related work of existing adversarial models and defensive methods. Then, Sect. 3 gives the details of the proposed EDLIC framework. After that, Sect. 4 shows extensive comparative results regarding the performance of deep learning models under different settings of adversarial environments, and discuss how to defend deep learning based image classification systems against existing adversaries. Finally, Sect. 5 concludes this paper.

2. Related Work

2.1 Security Threats against Deep Learning

2.1.1 Fast Gradient Sign Method

The perturbation of adversarial sample should be limited to undetectable by the naked eye, or not cause great damage to the legal sample, so usually, the norm is applied to limit the perturbation (different norms have different effects). Obviously, it is the most effective for the perturbation to occur at the direction of gradient, and because a reasonable hypothesis in the Fast Gradient Sign Method (FGSM) [12] is that the existing image data are stored in a discrete way, so the perturbation with a small order at the same direction of gradient will not have a significant impact on human vision. Therefore, FGSM gets the perturbation by shifting in the gradient direction of the original sample, as shown in (1), where ϵ is the perturbation factor, $J(\theta, \mathbf{x}, y)$ means the cost function, θ refers to the parameter of classification model and y is the true label of the original example x.

$$\mathbf{x}_{adv} = \mathbf{x} + \epsilon sign(\nabla_x J(\theta, \mathbf{x}, y)) \tag{1}$$

2.1.2 Carlini/Wagner Attacks

Carlini/Wagner (CW) is the attack model proposed by Carlini and Wagner [13]. The attack is not easily detected by limiting ℓ_∞, ℓ_2 and ℓ_0 norm, which is more effective than previous attack methods. Experiments have shown that distillation is unable to fully defend against CW attacks. The adversarial perturbation generated by the algorithm can be migrated from the unsecured network to the secured network to realize the black box attack. On the other hand, the CW with ℓ_2 makes the optimization problem much easier to be solved when searching for the best perturbation w, as show in (2), where $f(\cdot)$ refers to the objective function. Moreover, the optimal coefficient c is selected by performing 20 iterations of binary search algorithm.

$$\text{minimize} \|\frac{1}{2}(\tanh(w)+1) - x\|_2^2 + c \cdot f(\frac{1}{2}\tanh(w)+1) \tag{2}$$

2.1.3 DeepFool

DeepFool [14] solves the problem of choosing the perturbation coefficient in FGSM. DeepFool considers the case of linear decision function and obtains the optimal perturbation r^* by solving the optimization problem formulated in (3), where $f(\cdot)$ means the classifier, ω is a parameter set of the separating affine hyperplane satisfying $\mathcal{F} = \{\mathbf{x} : \omega^{\mathrm{T}}\mathbf{x} + b = 0\}$, and r represents the perturbation of classification results. Secondly, the attacking method against general nonlinear decision functions is implemented by multiple iterations of linearization procedure.

$$\begin{aligned} r^*(\mathbf{x}) &:= \arg\min_r \|r\|_2 \\ &\quad \text{subject to:} \quad f(\mathbf{x}+r) \neq f(\mathbf{x}) \\ &= -\frac{f(\mathbf{x})}{\|\omega\|_2^2}\omega. \end{aligned} \tag{3}$$

2.1.4 Jacobian-Based Saliency Map Method

Papernot et al. [15] proposed jacobian-based Saliency Map Method (JSMA). For an input example $\mathbf{x}$, the adversary aims to cause misclassification respect to $\mathbf{x}$, e.g., making the classifier output a designated wrong label t, which is different from the true label of $\mathbf{x}$. Let $P_m(\mathbf{x})$ denote the probability of misclassification to the label m. Thus, the objective of JSMA is to make an increase of $P_t(\mathbf{x})$ but a decrease of $P_j(\mathbf{x})$ ($j \neq t$) such that $t = \arg\max_j P_j(\mathbf{x})$. The Eq. (4) shows the formulation of increasing the ith input feature of an image $\mathbf{x}$ using the adversarial saliency map $S(\mathbf{x}, t)[i]$, where $P_t(\mathbf{x})$ is the probability of classifying $\mathbf{x}$ to the wrong label t, and $P_j(\mathbf{x})$ means that to the label j.

$$S(\mathbf{x}, t)[i] = \begin{cases} 0, & \text{if } \frac{\partial P_t(\mathbf{x})}{\partial \mathbf{x}_i} < 0 \text{ or } \sum_{j\neq t} \frac{\partial P_j(\mathbf{x})}{\partial \mathbf{x}_i} > 0 \\ \left(\frac{\partial P_t(\mathbf{x})}{\partial \mathbf{x}_i}\right) \left| \sum_{j\neq t} \frac{\partial P_j(\mathbf{x})}{\partial \mathbf{x}_i} \right|, & \text{otherwise} \end{cases} \tag{4}$$

2.1.5 Basic Iterative Method

Kurakin et al. [16] extended FGSM and proposed a basic iterative attack method (BIM). Specifically, BIM runs FGSM several times when a small time interval and observes the intermediate pixels after each time of trail. To ensure that all generated examples are located at the ϵ affinity of an original example, a hyperparameter α is adopted to control the value change of pixels at each step. Moreover, the number

of iterations is determined to be $\min(\epsilon+4, 1.25\epsilon)$ for the adversarial examples to reach the edge of the ϵ max-norm ball, as shown in (5), where $\mathbf{x}_0^{adv} = \mathbf{x}$, t denotes the true label of $\mathbf{x}$, and $Clip_{\mathbf{x},\epsilon}\{x'\}$ refers to the clipping of each pixel of $\mathbf{x}$.

$$\mathbf{x}_{k+1}^{adv} = Clip_{\mathbf{x},\epsilon}\{\mathbf{x}_k^{adv} + \alpha \cdot sign(\nabla_{\mathbf{x}} J(\mathbf{x}_{k+1}^{adv}, t))\} \tag{5}$$

2.1.6 Momentum Iterative Method

Dong et al. [17] proposed an Iterative attack Method (MIM) to generate adversarial examples $\mathbf{x}_{adv}$. This method can generate non-target adversarial samples from normal samples and reach the edge of ℓ_∞ norm. We can obtain the adversarial sample by solving the optimization constraint problem. FGSM has limitations in practical application, and it is easy to fall into extreme value and overfitting of the model, and its attack ability is limited. MIM combines momentum method with FGSM to achieve better attack effect, as shown in (6), where y is the ground-truth label of $\mathbf{x}$, and ϵ is the size of adversarial perturbation.

$$\arg\max_{\mathbf{x}_{adv}} J(\mathbf{x}_{adv}, y), \quad \text{s.t} \, \|\mathbf{x}_{adv} - \mathbf{x}\|_\infty \le \epsilon \tag{6}$$

2.1.7 Projected Gradient Descent

Madry et al. [18] extended FGSM and proposed a new attack method Projected gradient descent (PGD). Specifically, PGD utilizes the previous FGSM several times to find out the optimal adversarial examples by solving a large-scale constrained optimization problem, as shown in (7), where $\mathcal{S}$ means a projection range, and $L(\theta, \mathbf{x}, y)$ means the loss function with a parameter set θ of learning models. Experimental results validated that the local maxima of perturbance found by PGD all had similar loss values.

$$\mathbf{x}^{t+1} = \prod_{\mathbf{x}+\mathcal{S}} (\mathbf{x}^t + \alpha \cdot sgn(\nabla_{\mathbf{x}} L(\theta, \mathbf{x}, y))) \tag{7}$$

2.1.8 Elastic-Net Attack to DNNs

Chen et al. [19] proposed a new attack method called Elastic-net attack to DNNs (EAD), which extended the CW attack to enhance the transferability and to improve the success rate of this attack. Similar to CW, EAD uses the loss function $L(\mathbf{x}, t)$. However, unlike previous methods using the ℓ_1 or ℓ_2 regularization term, EAD adopts the elastic network regularization term, as shown in (8), where $L(\cdot)$ means the loss function, c is a regularization coefficient of $L(\cdot)$, and β refers to the penalty of the ℓ_1 term.

$$\begin{aligned} &\text{minimize}_{\mathbf{x}} \quad c \cdot L(\mathbf{x}, t) + \beta \|\mathbf{x} - \mathbf{x}_0\|_1 + \|\mathbf{x} - \mathbf{x}_0\|_2^2 \\ &\text{s.t.} \quad \mathbf{x} \in [0,1]^p \end{aligned} \tag{8}$$

2.1.9 Backward Pass Differentiable Approximation

Athalye at al. [20] proposed a new attack method called Backward Pass Differentiable Approximation (BPDA). Considering that many defensive techniques suffer from gradient confusion due to gradient fragmentation, random gradient or vanishing gradient, the BPDA attack can successfully overcome the gradient confusion. Specifically, the classifier model and the defending process are denoted as $f(\cdot)$ and $g(\cdot)$, respectively. Thus, a secure classifier model is obtained by calculating $S(\mathbf{x}) = f(g(\mathbf{x}))$. Assume that $g(\cdot)$ satisfies $g(\mathbf{x}) \approx \mathbf{x}$, then the derivative of $g(\mathbf{x})$ is approximately equal to 1, i.e., $\nabla_{\mathbf{x}} g(\mathbf{x}) \approx \nabla_{\mathbf{x}} \mathbf{x} = 1$. After that, the derivative of $S(\mathbf{x})$ can be obtained using (9).

$$\nabla_{\mathbf{x}} S(\mathbf{x}) \mid_{\mathbf{x}=\hat{\mathbf{x}}} = \nabla_{\mathbf{x}} f(g(\mathbf{x})) \mid_{\mathbf{x}=\hat{\mathbf{x}}} \approx \nabla_{\mathbf{x}} f(\mathbf{x}) \mid_{\mathbf{x}=g(\hat{\mathbf{x}})} \tag{9}$$

2.2 Defensive Technology of Deep Learning

2.2.1 Defensive Techniques during Training

As for defensive techniques, the main defense against the adversarial attack during training is data cleaning. Mainly by screening training data, compared with other defense methods is a more direct approach. For example, to determine whether a sample is an adversarial sample, add the candidate sample into the training data set, and then train. If the result of training is too different from the result of the original training data set, it can be judged that it is an adversarial sample and can be removed. Chen et al. [21] proposed KuafuDet method to solve malicious software poisoning on mobile phones. It consists of two phases to enhance the defensive performance against malware in an adversarial environment. The first is the offline mode, which extracts relevant features from the input data. The second is the online mode, which uses the features extracted before for adversarial training.

Another defense method is to improve the robustness of classifier so as to effectively reduce the influence of the adversary. Liu et al. [22] proposed a robust regression method that can effectively improve the security of classifiers in the face of poisoning attacks. Different from previous methods of this kind, which are based on strong assumptions of the feature matrix, this method reduces these assumptions and approximates the feature matrix to a low-rank matrix. Biggio et al. [23] proposed a Bagging method. Because they think the poisoning attack is a special kind of outliers, so they can use the Bagging ensembles for defense.

2.2.2 Defensive Techniques during Testing

Compared with the training process, the defense technology for the test process is to improve the safety of the deep neural network. Adversarial training can be adopted, aiming at generating adversarial samples actively during training, and then extracting adversarial samples during testing to improve the security of the model. Moreover, we can use the fast gradient sign method to generate adversarial samples at low cost and high efficiency and give the adversarial samples

correct label. Specifically, given an adversarial sample of a car, we use it to trick the model into thinking it's a bird, and then we tell the model that the correct label for the photo is a car. Tramel et al. [24] proposed a new method of ensemble adversarial training. They separated the pre-generated adversarial samples from the training model to ensure that it was difficult to evaluate the adversarial samples during the training process. In addition, adversarial samples are generated by independent models, which provides a new idea for protecting classifiers from black box attacks. Then, ensemble adversarial training is evaluated in MNIST and ImageNet, and the results show that this method significantly enhanced the robustness of classifier to the adversarial samples.

Another compression-based defensive technique can effectively compress the noise in the sample, hence prevent the perturbation of the adversarial models. Das et al. [25] proposed a fast and effective defense method based on JPEG compression. At the same time, in order to reduce the impact of compression on images, they retrain the classification model with the compression samples. Experiment results show that this JPEG-based defense technology can effectively eliminate the black box-attacks and gray-box attacks.

2.3 Security Assessment in Image Classification

With the wide application of intelligent system, its security problem is increasingly serious. As the vulnerability of machine learning models cannot be ignored, there are some existing studies on evaluating the robustness of intelligent systems in adversarial environments. Baggio et al. [26] proposed an evaluation system to comprehensively analyze the security of the pattern classifier at the design stage. Furthermore, they carried out evaluation work in the spam classification model, biometric recognition model and intrusion detection system. The results show that the evaluation system can help the security staff to understand the behavior of the pattern classifier in adversarial environment and design a classification model with higher security. Weng et al. [27] proposed a new robust evaluation metric $CLEVER$ for the lack of security evaluation mechanism of classification model. More specifically, extreme value theory is used to design the metric $CLEVER$ that can be applied to large-scale neural networks. When evaluating the adversarial sample, the $CLEVER$ metric is consistent with ℓ_2 and ℓ_∞ norms, and the classification model using defense technology has a higher $CLEVER$ value.

There is still a lot of research space in the serious problem of evaluating robustness and security of the intelligent classification model in adversarial environment. Therefore, we study the influence of adversary model and defense technology on the robustness of image classification model, and propose a set of evaluation metrics.

3. The Proposed EDLIC Framework

3.1 Security Assessment Framework Overview

Generally speaking, the goal of examining the security of an intelligent system is to investigate (1) its performance degradation when facing adversaries and (2) its robustness against these security threats by adopting proper defensive techniques. In this paper, we propose a security evaluation framework named EDLIC to comprehensively assess the security of a given deep learning based image classification system in the adversarial environment, as illustrated in Fig. 1. Specifically, the proposed EDLIC framework conducts quantitative performance evaluation towards the deep learning system from two aspects, i.e., the performance when being attacked by adversarial examples, which are generated via different attacking models, and that after adopting some defensive techniques.

3.1.1 Performance Evaluation Design for Adversarial Examples

To evaluating the performance of deep learning based image classification systems when being attacked, we first select a set of alternative adversarial models that generate adversarial examples. Typically, existing adversaries introduce noise data into original examples, e.g., dog pictures or digital images. By doing so, the resulting adversarial examples (e.g., the faked dog pictures or digital images) are regarded as the original category by human-beings but another one by deep learning-based image classification. After that, the EDLIC framework assesses the selected adversarial models by applying them to attack against deep learning algorithms and examining the extent to which their learning performance is degraded by adversarial examples.

3.1.2 Performance Evaluation Design for Defensive Techniques

After generating adversarial examples via different attacking models, the EDLIC framework investigates the effectiveness of state-of-the-art defensive techniques that protecting deep learning-based image classification systems against these adversarial examples. Specifically, the basic idea of evaluating defensive techniques is described as follows: Firstly, we define a transformation function adopting training examples (including normal and adversarial ones) as the input data. Such a function is regarded as a specific defensive method. Then, we utilize the function to transform original examples, and we term the results as *defensive examples*. After that, we feed the defensive examples into deep learning-based image classification systems to examine the performance improvement by using the defensive method to defend against adversaries. Finally, we compare the performance of different defensive methods by introducing different transformation functions.

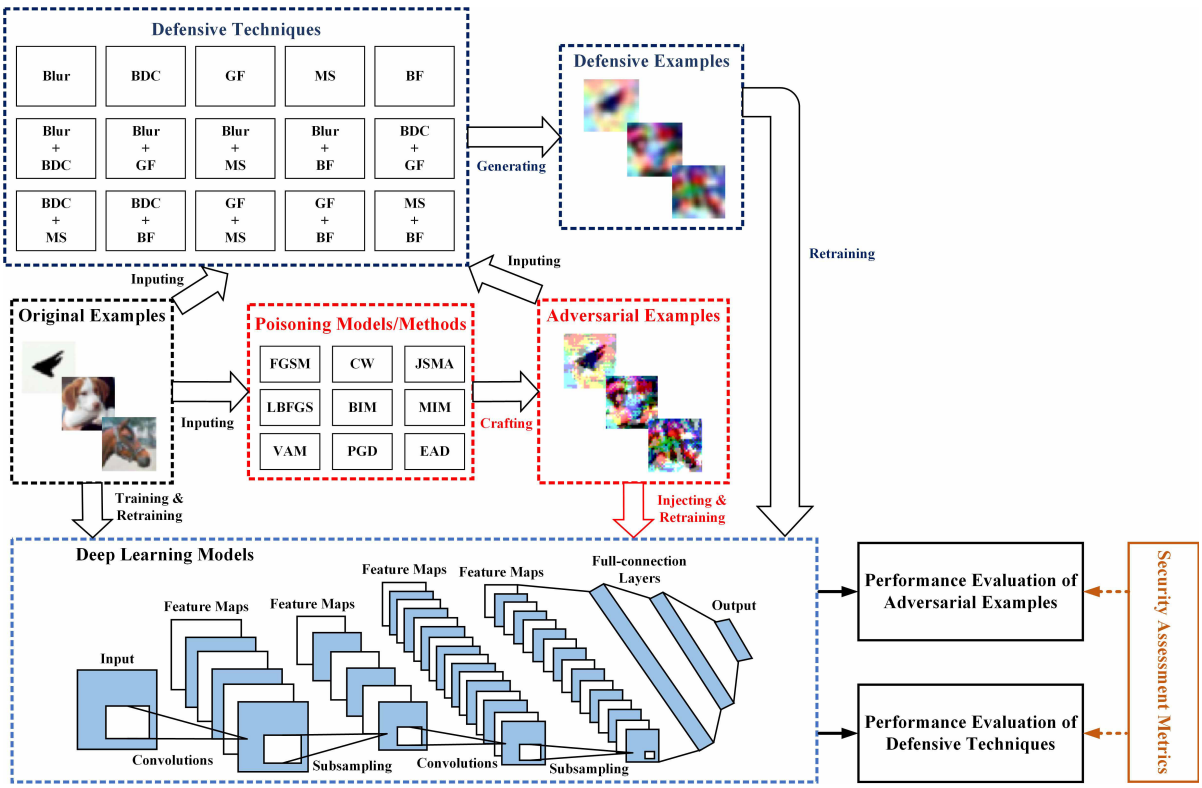


Fig. 1 The overall framework of EDLIC.

3.2 Security Assessment Metrics

To quantitatively evaluate the performance of adversarial examples and defensive techniques, we further define some security assessment metrics for the proposed EDLIC framework. Specifically, we introduce a variable (K) for evaluating the distortion rate between an adversarial example and its corresponding original one. Moreover, we propose two assessment metrics that are related to a target model with the presence of adversaries: (1) model accuracy (ACC) and (2) success rate of adversaries R_{ad}, which means the ratio of the number of examples that are classified into wrong categories specified by adversaries to that of all original ones.

Furthermore, we introduce δ_{OD} to measure predictive deviation between original and defensive examples and δ_{AD} between adversarial and responding defensive examples, where

$$\delta = \frac{\sum_{i=1}^{n} \| V(S_i) - V(DS_i) \|_1}{n}. \tag{10}$$

In the Eq. (10), $V(S_i)$ refers to the predicting probabilities of original/adversarial examples with respect to different categories, $V(DS_i)$ represents the predicting results of corresponding defensive examples, n means the number of data points.

Table 1 shows a summary of the above security assessment metrics.

Table 1 Summary of security assessment metrics.

Metric	Explanation
K	The dissimilarity between an adversarial example and its corresponding original one.
ACC	The ratio of the number of correctly classified examples to that of all original ones.
R_{ad}	The ratio of the number of examples that are classified into wrong categories specified by adversaries to that of all original ones.
δ_{OD}	The prediction deviation between original and corresponding defensive examples.
δ_{AD}	The prediction deviation between adversarial and corresponding defensive examples.

3.3 Methodology of Evaluating the Performance of Adversarial Models

Generally speaking, different adversarial methods compromising a target deep learning system shall generate adversarial examples with different features, so does a specific attacking model running over different training data sets. Hence, we propose to examine the quantitative performance of generated adversarial samples towards various image classification systems using the aforementioned assessment metrics as a rational way of evaluating the overall performance of different attacking models.

Specifically, at the phase of adversarial example generation, genuine image data are fed into an attack model. After

that, adversaries capture the data characteristics of the target classifier directly (under the white-box assumption) or tentatively (under the black-box assumption). Finally, adversarial examples are generated via an attacking model based on the captured data distribution.

Once obtaining the adversarial examples, we feed them into the original deep learning model and observe the performance of attack samples. Hence, at the phase of adversarial example evaluation, we propose to evaluate the performance of adversarial methods from two aspects: (a) the degree of training example distortion, and (b) the success rate of target adversarial sample. Basically, a better adversarial method should enjoy a smaller degree of example distortion and a much large value of the attack success rate, fooling the classification of deep learning based image misclassify the adversarial sample as the attack target.

3.4 Methodology of Evaluating the Performance of Defensive Techniques

Since different filters have different effects, in order to evaluate which type of filter can better defend against the eight common attacks nowadays, we select the following five defensive methods for performance evaluation: Blur, Bit-depth compression, Gaussian filtering, Mean smoothing, and Bilateral filtering. Before describing the workflow of evaluating the performance of defensive techniques, we briefly introduce the basic ideas of these five methods.

Blur: We use a low pass filter to smooth the image. More specifically, the goal of a low pass filter is to reduce the rate of image change. As a result, we replace each pixel with the average of the pixels around that pixel, so that we can smooth out the areas that have changed significantly.

Bit-Depth Compression (BDC): Neural networks are differentiable models that assume the input space is continuous, but digital computers only support discrete as an approximation of continuous natural data. In view of a standard digital image is represented by an array of pixels, each of which is usually represented as a specific color number, and there is much space available to an attacker. In this condition, Bit-depth compression reduces the amount of search space in an image, which is available to an attacker, and thus reduces the success rate of attack without compromising the accuracy of the classifier.

Gaussian Filtering (GF): In some cases, more attention needs to be paid to certain peripheral pixels of the pixel, so we can recalculate the values of peripheral points by assign weights, which can be solved by the Gaussian weight scheme. Compared with low-pass filtering, the weight of each pixel is the same, that is, the filter is linear, while the weight of the pixel in the Gaussian filter is proportional to its distance from the center pixel.

Mean Smoothing (MS): Mean smoothing is a widely used technique to reduce image noise, more specifically, the median value of pixels in the convolution box is used to replace the central value to achieve the purpose of denoising. In reality, the filter runs a sliding window on each pixel of the image, with the center pixel replaced by the median of the adjacent pixels within the window. Rather than actually reducing the number of pixels in the image, it propagates pixel values on neighboring pixels, and the median filter basically compresses the sample by making adjacent pixels more similar.

Bilateral Filtering (BF): Bilateral filtering not only considers the spatial relation of the image but also the gray relation of the image. At the same time, the Gaussian weight of space and Gaussian weight of gray similarity are used to ensure that the boundary is not blurred. Hence, it consists of two functions, one of which is the set space distance determining the filter coefficient and the other is the pixel difference.

For the sake of fairness, we compare the above five defensive methods using four indicators: ACC of learning models with or without defensive techniques in different adversarial environments, R_{ad} with or without defensive techniques, δ_{OD} and δ_{AD}.

4. Comparative Results and Analysis

4.1 Experimental Setup

In this section, we selected two well-known image classification datasets, i.e., MNIST[†], FASHION-MNIST[††], SVHN[†††], CIFAR-10[††††] and CIFAR-100[†††††] for comparative experiments. Specifically, the MNIST dataset consists of 70000 black-and-white images of hand-written numbers 0 to 9, and FASHION-MNIST consists of 70000 grayscale images of 10 fashion items. Moreover, the SVHN dataset includes 630420 images of real house number from Google Street View and a total of 10 categories. On the other hand, the CIFAR-10 dataset consists of 60000 RGB color images with 32*32 pixels and a total of 10 categories. The CIFAR-100 dataset consists of 60000 RGB color images in 100 classes. To establish a good baseline of performance comparison, we selected different deep learning models to handle different datasets, as shown in Table 2, where CNN was

Table 2 The accuracy of target models over different benchmark datasets.

Dataset	Model	Accuracy
MNIST	CNN Model	99.37%
FASHION-MNIST	CNN Model	90.88%
SVHN	TFlearn CNN Model	89.33%
CIFAR-10	CNN Model	76.34%
CIFAR-100	Resnet-20 Model	70.13%

[†]The MNIST dataset: http://yann.lecun.com/exdb/mnist/

[††]The FASHION-MNIST dataset: https://github.com/zalandoresearch/fashion-mnist

[†††]The SVHN dataset: http://ufldl.stanford.edu/housenumbers/

[††††]The CIFAR-10 dataset: https://www.cs.toronto.edu/~kriz/cifar.html

[†††††]The CIFAR-100 dataset: https://www.cs.toronto.edu/~kriz/cifar.html

Fig. 2 Illustration of the adversarial examples crafted by different adversarial methods, where the first column shows original images from MNIST, FASHION-MNIST, SVHN, CIFAR-10 and CIFAR-100 datasets in a top-down manner, and remaining columns present corresponding adversarial images from eight attacking methods FGSM, CW, JSMA, LBFGS, BIM, MIM, PGD and EAD, respectively.

chosen to handle the MNIST, FASHION-MNIST, CIFAR-10 datasets, TFlearn CNN models was chosen to handle the SVHN dataset, and Resnet-20 model was chosen to handle the CIFAR-100 dataset.

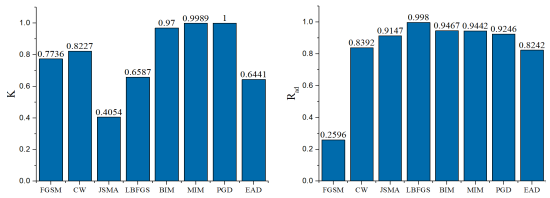
To evaluate the performance of adversarial examples crafted by different adversarial methods, we selected 8 typical methods as follows: FGSM, CW, JSMA, LBFGS, BIM, MIM, EAD, and PGD. Taking four images, each of which is respectively selected from MNIST, FASHION-MNIST, SVHN, CIFAR-10 and CIFAR-100 datasets, as an example, the adversarial examples crafted by these adversarial methods are shown in Fig. 2. During the experiments, we installed TensorFlow and implemented these adversarial methods using the *cleverhans* library, which was compiled and tested using python3.

Furthermore, we implemented 5 defensive techniques as follows: Blur, BDC, GF, MS, and BF. Specifically, we used CV2.BoxFilter in OpenCV to implement the Blur method. Regarding the implementation of BDC, we specified the range of input and output values in $[0, 1]$, and the input value was multiplied by 2^i-1 to compress the bit depth of the ith bit. For outputs, the corresponding value was adjusted to $[0, 1]$ by dividing $2^i - 1$. Regarding the implementation of GF, MS and BF, we utilized the GaussianBlur, medianBlur and bilateralFilter functions that are defined in the CV2 library of OpenCV.

To validate the effectiveness of the proposed EDLIC, we adopted two evaluating metrics K and R_{ad} to evaluate the performance of adversarial examples and four metrics ACC, R_{ad}, δ_{OD} and δ_{AD} to defensive methods.

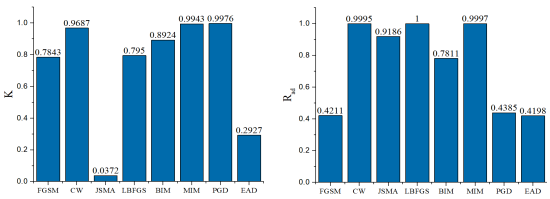
4.2 Comparative Results and Analysis Regarding the Performance of Adversarial Samples

The relation between K and R_{ad} over MNIST dataset is clear at a glance, as shown in Fig. 3. There is no linear correlational relationship among K and R_{ad}, e.g., FGSM, introduce a significant difference between original examples and corresponding adversarial ones (K = 77.36% for FGSM) but



(a) K (b) R_{ad}

Fig. 3 Comparative results of evaluating different adversarial methods against the MNIST dataset.



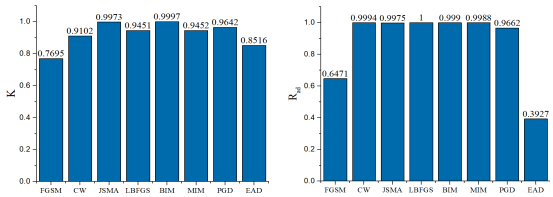
(a) K (b) R_{ad}

Fig. 4 Comparative results of evaluating different adversarial methods against the FASHION-MNIST dataset.

a small value of R_{ad} (R_{ad} = 25.96% for FGSM). On the contrary, some attacks, e.g., LBFGS, obtain a large value of R_{ad} (R_{ad} = 99.80% for LBFGS) but a small value of K (K = 65.87% for LBFGS). Regarding PGD, both k and R_{ad} indicate the considerable performance of the adversarial method. In addition, JSMA, LBFGS, BIM, MIM, and PGD achieve over 91.47% in terms of the R_{ad}, where LBFGS gets the best result (up to 99.8%). In terms of K, the result of JSMA is the most satisfying. Therefore, we recommend the JSMA method, which has a good effect of attack, low distortion rate, as the best method to attack against the MNIST dataset. Such a conclusion is helpful for security researchers and practitioners to design targeted and effective countermeasures.

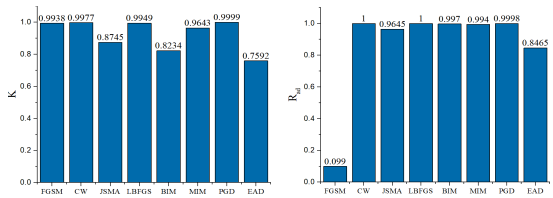
The relation between K and R_{ad} over FASHION-MNIST dataset is clear at a glance, as shown in Fig. 4. FGSM and PGD attacks are not satisfactory, since they introduce a significant difference between original examples and corresponding adversarial ones (K = 78.43% for FGSM, K = 99.76% for PGD) but a small value of R_{ad} (R_{ad} = 42.11% for FGSM, R_{ad} = 43.85% for PGD). In addition, CW, JSMA, LBFGS and MIM achieve over 91.86% in terms of the R_{ad}, where LBFGS gets the best result (up to 100%). In terms of K, the result of JSMA is 3.72%, which is smaller than the results of other methods. By comparing and analyzing, we recommend the JSMA method, which has the good-attacking effect and smallest distortion rate, as the best method to attack against the FASHION-MNIST dataset.

The relation between K and R_{ad} over SVHN dataset is clear at a glance, as shown in Fig. 5. EAD has a large value of K (K = 85.16% for EAD) but a small value of R_{ad} (R_{ad} = 39.27% for EAD). In addition, CW, JSMA, LBFGS, BIM, MIM, and PGD achieve over 96.62% in terms of the R_{ad}, where LBFGS gets the best result (up to 100%). In general, we recommend the CW method, which has good-



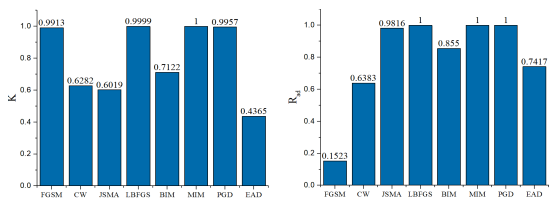
(a) K (b) R_{ad}

Fig. 5 Comparative results of evaluating different adversarial methods against the SVHN dataset.



(a) K (b) R_{ad}

Fig. 6 Comparative results of evaluating different adversarial methods against the CIFAR-10 dataset.



(a) K (b) R_{ad}

Fig. 7 Comparative results of evaluating different adversarial methods against the CIFAR-100 dataset.

attacking effect and a ralatvely small value of distortion rate. This conclusion helps security researchers to be on the alert for CW attack methods over SVHN dataset.

The comparative results of evaluating different adversarial methods against the CIFAR-10 dataset are shown in Fig. 6. There is also no linear correlational relationship among K and R_{ad}. In addition, the values of R_{ad} with respect to CW, JSMA, LBFGS, BIM, MIM, and PGD are all over 96.45%. However, the K of attacks over CIFAR-10 dataset are not low. We recommend JSMA and BIM, which achieves large results for R_{ad} and a relatively small result for K, as the most efficient adversarial method for use to attack against the CIFAR-10 dataset.

The relation between K and R_{ad} over CIFAR-100 dataset is clear at a glance, as shown in Fig. 7. FGSM attack is not satisfactory, whic introduce a significant difference between original examples and corresponding adversarial ones (K = 99.13% for FGSM) but a small value of R_{ad} (R_{ad} = 15.23% for FGSM). On the contrary, some attacks, e.g., MIM, obtain a large value of both R_{ad} and K. In addition, JSMA, LBFGS, MIM, PGD achieve over 98.16% in terms of the R_{ad}. By comparing and analyzing, we recommend the JSMA method, which has a good attacking effect, low distortion rate (K = 60.19% for JSMA), as the best method to attack against the CIFAR-100 dataset.

4.3 Comparative Results and Analysis Regarding the Performance of Defensive Techniques

4.3.1 Comparative Results and Analysis Regarding R_{ad} and ACC

The comparative results of the performance of different defensive techniques in terms of R_{ad} over MNIST dataset are shown in Table 3. Among the five defensive techniques Blur, BDC, GF, MS and BF, the BDC can achieve the best performance in protecting deep learning models against the above adversarial methods because the BGD method can minimize R_{ad} of different attacks except for JSMA. More specifically, BDC achieves the best results of R_{ad} from 0.2596 to 0.0053, 0.8392 to 0.0034, 0.998 to 0.108, 0.9467 to 0.0033, 0.9442 to 0.0043, 0.9246 to 0.0025 and 0.8242 to 0.2099 for FGSM, CW, LBFGS, BIM, MIM, PGD and EAD, respectively. The comparative results of the performance of different defensive techniques in terms of ACC over MNIST dataset are shown in Table 4. Among the five defensive methods Blur, BDC, GF, MS and BF, the BDC can maximize the ACC of deep learning models except for JSMA. As we can see, BDC achieves the best results of the ACC over 83.1% for FGSM, CW, LBFGS, BIM and MIM. The conclusion is consistent with the above analysis about R_{ad}. In summary, for the MNIST dataset, if only one single defensive method is used, then we recommend BDC as the most suitable method for protecting deep learning models against various attacks. On the other hand, BDC can be integrated with other defensive methods to offer more powerful defensive methods that are able to reduce R_{ad} and enhance ACC with the presence of adversarial examples.

The results of R_{ad} over FASHION-MNIST dataset are shown in Table 3. The BDC can protect deep learning models against most adversarial methods, which is consistent with the conclusion of MNIST dataset. Therefore we recommend BDC to defend against FGSM/LBFGS/BIM/MIM/PGD/EAD, Blur to CW, and MS to JSMA. The results of ACC over FASHION-MNIST dataset are shown in Table 4. Similarly, the BDC can defend against most attacks (e.g., FGSM, LBFGS, BIM, MIM, PGD and EAD). In addition, MS can effectively reduce the perturbation of JSMA attack, and BF can perform well in CW attack.

The results of R_{ad} over SVHN dataset are shown in Table 3. The BDC and MS can achieve the best performance in protecting deep learning models against the above adversarial methods because these methods can minimize R_{ad} of different attacks. The results of ACC over SVHN dataset are shown in Table 4. Among the five defensive methods, the best performance method varies from the adversarial environment, e.g., MS achieves the best results of ACC from 0.02% to 55.84%, 0.23% to 83.6% for CW and JSMA attacks. In summary, for the SVHN dataset, we recommend BDC and MS, as the most suitable method for protecting deep learning models against various attack attacks.

Table 3 Comparative results of the performance of different defensive techniques in terms of R_{ad}.

Dataset	Defensive Techniques	FGSM	CW	JSMA	LBFGS	BIM	MIM	PGD	EAD
MNIST	None	0.2596	0.8392	0.9147	0.998	0.9467	0.9442	0.9246	0.8242
	Blur	0.1754	0.1195	0.2129	0.342	0.8007	0.7941	0.7029	0.406
	BDC	**0.0053**	**0.0034**	0.5082	**0.108**	**0.0033**	**0.0043**	**0.0025**	**0.2099**
	GF	0.1441	0.0974	0.1555	0.273	0.7751	0.7899	0.6567	0.3328
	MS	0.1861	0.2578	**0.0169**	0.586	0.7868	0.7885	0.6949	0.4668
	BF	0.2344	0.1038	0.2768	0.911	0.9198	0.9066	0.8711	0.6739
FASHION-MNIST	None	0.4211	0.9995	0.9186	1	0.7811	0.9997	0.4385	0.4198
	Blur	0.2394	**0.0344**	0.11	0.249	0.4633	0.7628	0.1367	0.265
	BDC	**0.0696**	0.0822	0.2443	**0.1083**	**0.1128**	**0.338**	**0.0447**	**0.2576**
	GF	0.2213	0.0509	0.129	0.2262	0.4384	0.7446	0.1327	0.2592
	MS	0.2986	0.0537	**0.0075**	0.2896	0.5418	0.8009	0.2631	0.2841
	BF	0.3328	0.0611	0.1737	0.6298	0.673	0.9811	0.3062	0.3675
SVHN	None	0.6471	0.9994	0.9975	1	0.999	0.9988	0.9662	0.3927
	Blur	0.5989	0.2946	0.1044	0.7218	0.9422	0.9172	0.7268	0.2986
	BDC	**0.3985**	0.2345	0.111	**0.4442**	**0.7288**	**0.7075**	**0.504**	**0.2488**
	GF	0.6095	0.2562	0.0873	0.6561	0.9342	0.9083	0.707	0.3072
	MS	0.5597	**0.2291**	**0.0256**	0.6534	0.9461	0.9239	0.721	0.3018
	BF	0.6448	0.6389	0.2175	0.9697	0.9951	0.9946	0.9202	0.3695
CIFAR-10	None	0.099	1	0.9645	1	0.997	0.994	0.9998	0.8465
	Blur	0.1041	0.065	0.4265	0.0397	0.3798	0.434	0.1346	0.0447
	BDC	**0.0797**	0.1125	0.3215	0.1085	0.5811	0.718	0.4726	**0.0157**
	GF	0.0985	0.5353	0.1256	0.0424	**0.2028**	0.3476	**0.0914**	0.0572
	MS	0.119	**0.0464**	**0.0152**	**0.0362**	0.333	**0.2916**	0.1478	0.0446
	BF	0.1182	0.0983	0.6526	0.058	0.929	0.8426	0.5615	0.0852
CIFAR-100	None	0.1523	0.6383	0.9816	1	0.855	1	1	0.7417
	Blur	0.0326	0.0117	0.1265	0.0082	**0.0284**	0.0567	0.0391	0.0133
	BDC	**0.0032**	0.0154	0.0896	**0.0016**	0.0325	**0.0436**	**0.0312**	0.0117
	GF	0.0259	0.0195	0.1567	0.0256	0.063	0.1982	0.1172	0.015
	MS	0.0345	**0.0057**	**0.0325**	0.0137	0.0354	0.0634	0.0508	**0.0067**
	BF	0.1386	0.0083	0.2368	0.1095	0.2175	0.6531	0.6406	0.0133

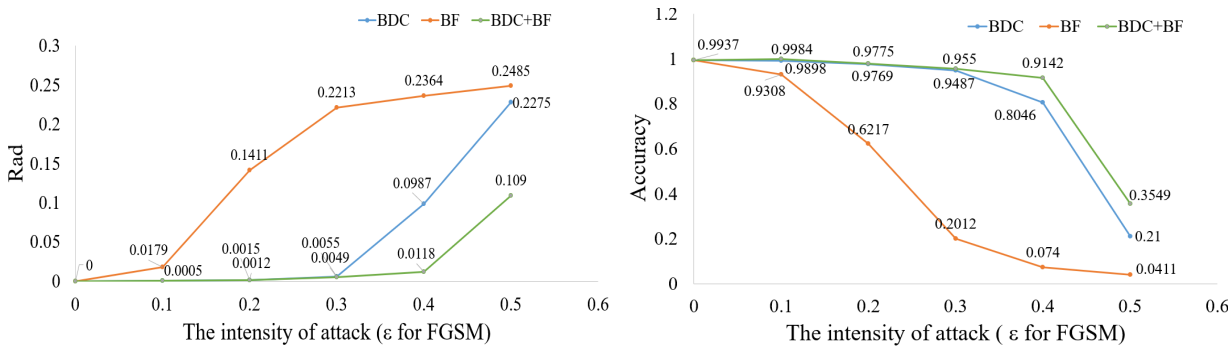
The comparative results of the performance of different defensive techniques in terms of R_{ad} over CIFAR-10 dataset are shown in Table 3. We can see that the selection of proper defensive methods varies from target attack methods. Specifically, we recommend BDC to defend against FGSM/EAD, GF to BIM/PGD and MS to CW/JSMA/LBFGS/MIM. The comparative results of the performance of different defensive techniques in terms of *ACC* over CIFAR-10 dataset are shown in Table 4. Typically, we can draw a similar conclusion that the selection of proper defensive methods also varies from target attack methods. For example, we recommend GF to defend against FGSM/BIM/MIM, BF to CW/LBFGS, MS to JSMA and Blur to PGD. Based on the above analysis, for the CIFAR-10 dataset, if only one single defensive method is used, then we recommend GF or MS as the most suitable method to protecting deep learning models against various attack attacks. On the other hand, GF or MS can also be integrated with other defensive methods to reduce R_{ad} and enhance *ACC* when resisting against adversarial examples crafted by different attack methods.

The results of R_{ad} over CIFAR-100 dataset are shown in Table 3. The selection of best performance defensive techniques about reducing R_{ad} varies from the adversarial environment, therefore we recommend Blur to defend against BIM, BDC to FGSM/LBFGS/MIM/PGD, and MS to CW/JSMA/EAD. The results of *ACC* over CIFAR-100 dataset are shown in Table 4. Similarly, the selection of best performance defensive techniques to enhancing *ACC* varies from an adversarial environment, e.g., GF perform well in JSMA attack, MS perform well in FGSM, LBFGS, MIM and PGD attacks, BF perform well in CW, BIM and EAD attacks.

Regarding combined defensive techniques, most of them have better performance than single defensive methods, e.g., BDC+BF over MNIST dataset in the FGSM adversarial environment, as shown in Fig. 8. Specifically, in case of increasing FGSM intensity, the combined defensive technique can effectively slow down the growth of the success rate of attack and maintain the accuracy of the classification model than single defensive methods. When the perturbation coefficient of FGSM is 0.4, BDC+BF reduce the R_{ad} of attack to 0.0118, which is effective than BDC and BF; BDC+BF improve the *ACC* of the model to 0.9142, which is much large than BDC and BD. The result is rational because it is not clear which attack method an adversary will choose in the real world. Hence, we can integrate different defensive techniques to make use of the advantages of different methods, when protecting deep learning models against a variety of attacks.

Table 4 Comparative results of the performance of different defensive techniques in terms of *ACC*.

Dataset	Defensive Techniques	FGSM	CW	JSMA	LBFGS	BIM	MIM	PGD	EAD
MNIST	None	0.1812	0.1602	0.0564	0.002	0.0435	0.0491	0.0716	0.1572
	Blur	0.2829	0.7137	0.4533	0.359	0.1453	0.1507	0.2198	0.29
	BDC	**0.95**	**0.9676**	0.3943	**0.831**	**0.9826**	**0.9804**	**0.986**	**0.6738**
	GF	0.2688	0.6478	0.4368	0.327	0.1583	0.1566	0.2371	0.3122
	MS	0.1744	0.4921	**0.8777**	0.226	0.1317	0.1116	0.1847	0.2643
	BF	0.2392	0.8341	0.5575	0.081	0.0669	0.0798	0.1212	0.219
FASHION-MNIST	None	0.2236	0.0005	0.0387	0	0.1381	0	0.477	0.5127
	Blur	0.3829	0.7265	0.6309	0.5715	0.3918	0.1313	0.6659	0.4999
	BDC	**0.6058**	0.6496	0.5118	**0.6981**	**0.7079**	**0.5319**	**0.7523**	**0.5527**
	GF	0.3719	0.6929	0.5806	0.5717	0.4099	0.1416	0.6506	0.4905
	MS	0.2982	0.7344	**0.7895**	0.5418	0.3088	0.0628	0.5437	0.5058
	BF	0.3017	**0.7496**	0.6337	0.2822	0.2309	0.0055	0.5732	0.5241
SVHN	None	0.0214	0.0002	0.0023	0	0.0005	0.0012	0.0318	0.5464
	Blur	0.0647	0.515	0.6894	0.2449	0.0491	0.0736	0.245	**0.5589**
	BDC	**0.097**	0.3085	0.5102	0.2448	**0.1045**	**0.1309**	0.247	0.3965
	GF	0.068	0.5476	0.7175	**0.298**	0.0553	0.0816	**0.2615**	0.5569
	MS	0.0574	**0.5585**	**0.836**	0.295	0.0453	0.066	0.2439	0.5523
	BF	0.0304	0.2719	0.6106	0.0284	0.004	0.0052	0.0759	0.5527
CIFAR-10	None	0.1122	0	0	0	0	0	0	0.6813
	Blur	0.2	0.475	0.2495	0.4481	0.2156	0.1194	**0.3285**	0.3935
	BDC	0.1466	0.2327	0.3105	0.2838	0.0828	0.0426	0.1246	0.3096
	GF	**0.2367**	0.3398	0.4589	0.3515	**0.2554**	**0.1646**	0.3152	0.3224
	MS	0.1189	0.4686	**0.674**	0.5096	0.2246	0.0954	0.2893	0.4616
	BF	0.1839	**0.5166**	0.3546	**0.6962**	0.0243	0.0339	0.1955	0.6527
CIFAR-100	None	0.0533	0.1983	0.045	0	0.0554	0	0	0.1467
	Blur	0.1427	0.19	0.156	0.16	0.1856	0.1641	0.1875	0.1917
	BDC	0.0147	0.0767	0.0168	0.0226	0.0462	0.0435	0.0547	0.0767
	GF	0.1171	0.2317	**0.3569**	0.1438	0.1927	0.1564	0.1953	0.2267
	MS	**0.1451**	0.2417	0.1256	**0.1662**	0.2016	**0.1756**	**0.1992**	0.2456
	BF	0.1157	**0.3033**	0.2684	0.0863	**0.2213**	0.1063	0.1523	**0.2883**



(a) Comparative the R_{ad} (b) Comparative the ACC

Fig. 8 The R_{ad} and ACC of combined defensive technique and single defensive techniques in FGSM adversarial environment over MNIST dataset.

4.3.2 Comparative Results and Analysis Regarding δ_{OD}

It can be seen from Fig. 9 that the predicting deviation δ_{OD} between original and corresponding defensive examples is very small, e.g., in MNIST dataset, 90% of the δ_{OD} values with Blur, BDC, GF, MS and BF defensive techniques are small than 0.1625, 0.0804, 0.0918, 0.1334 and 0.1574, respectively. These results demonstrate that using the aforementioned defensive techniques in learning models will not affect the accuracy of original examples. These results also suggest that sample with a large value of prediction deviation can be regarded as an adversarial sample.

4.3.3 Comparative Results and Analysis Regarding δ_{AD}

Table 5 shows the comparative results of δ_{AD} under different attacking and defensive methods over the MNIST, FASHION-MNIST, SVHN, CIFAR-10 and CIFAR-100 datasets. In contrast to δ_{OD}, the comparative results of

Table 5 Comparative results of the prediction deviation δ_{AD}.

Dataset	Defensive Techniques	FGSM	CW	JSMA	LBFGS	BIM	MIM	PGD	EAD
MNIST	Blur	0.2199	0.3672	0.3069	0.571	0.5111	0.4474	0.4866	0.4076
	BDC	**1.1353**	**0.5558**	0.1286	**0.9365**	**1.656**	**1.5935**	**1.6041**	**0.6992**
	GF	0.2401	0.409	0.3613	0.6229	0.5721	0.4978	0.5556	0.4565
	MS	0.2323	0.4921	**0.8637**	0.4877	0.4807	0.4377	0.4765	0.4096
	BF	0.0928	0.1624	0.1343	0.2179	0.136	0.1378	0.1403	0.1472
FASHION-MNIST	Blur	0.6269	0.7348	0.5765	0.9848	0.7728	0.5366	0.8471	0.6069
	BDC	**1.1184**	**0.9661**	0.5297	**1.3494**	**1.4894**	**1.2945**	**1.2717**	0.5915
	GF	0.648	0.7413	0.5883	0.9984	0.8049	0.5809	0.8689	**0.639**
	MS	0.5554	0.8198	**1.672**	0.9812	0.5809	0.4391	0.6453	0.5402
	BF	0.2425	0.4054	0.3224	0.5256	0.2748	0.1518	0.3901	0.2358
SVHN	Blur	0.3167	0.4459	0.4183	0.781	0.3332	0.3309	0.5733	0.3328
	BDC	**0.6866**	**0.785**	**0.8189**	0.7211	**0.6245**	**0.6709**	**0.9195**	**0.614**
	GF	0.3159	0.463	0.4281	**0.8368**	0.3594	0.3617	0.6173	0.3196
	MS	0.3505	0.5013	0.7761	0.8427	0.2935	0.3247	0.5923	0.348
	BF	0.0901	0.1362	0.1464	0.2284	0.0602	0.0589	0.1396	0.0944
CIFAR-10	Blur	0.6022	0.7548	0.7462	1.8299	0.1225	0.2943	0.7469	0.6398
	BDC	0.5547	**1.1589**	0.6528	1.7984	0.3232	0.0493	0.4798	0.8893
	GF	0.8399	1.0614	0.5423	**1.8758**	**1.0429**	1.0083	**1.6784**	**0.964**
	MS	**0.9519**	0.9772	**1.0268**	1.8612	0.7378	**1.1687**	1.3957	0.8158
	BF	0.8094	0.8682	0.9452	1.8601	0.2571	0.5536	0.9905	0.8096
CIFAR-100	Blur	**1.1711**	1.0026	0.7658	1.8176	1.4019	**1.8758**	**1.9495**	1.0288
	BDC	1.1488	**1.1938**	0.9155	**1.8516**	**1.4435**	1.8711	1.9445	**1.2093**
	GF	1.1206	0.9276	0.9564	1.796	1.3432	1.7819	1.8833	0.9612
	MS	1.1028	0.9055	**1.3065**	1.8021	1.3682	1.8622	**1.9495**	0.9487
	BF	0.8969	0.7445	0.8468	1.7273	1.1401	1.4288	1.5121	0.7899

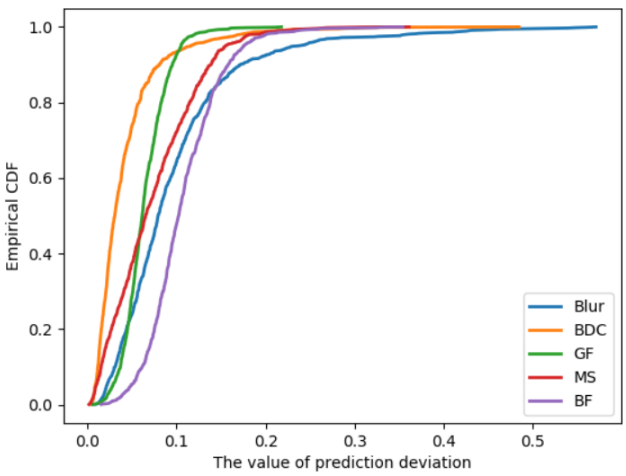


Fig. 9 Empirical CDF of δ_{OD} with Blur, BDC, GF, MS and BF defensive techniques over MNIST dataset.

δ_{AD} show that the prediction deviations between adversarial samples and corresponding defensive examples are very large. Moreover, the results of δ_{AD} under different datasets are significantly different. Regarding MNIST FASHION-MNIST and SVHN datasets, the BDC method is more effective than other defensive techniques, which is consistent with the results of *ACC*. For the CIFAR-10 dataset, GF and MS defensive techniques perform well in terms of δ_{AD}. Regarding the CIFAR-100 dataset, the Blur and BDC methods are better than other ones in terms of δ_{AD}.

4.4 Discussion and Suggestion

For adversarial models, ℓ_0 attacks (e.g., JSMA) have a small distortion rate, which is consistent with the expected conjecture. Since this type of attacks is targeted to add salt-like noise to the image, it is rational that the distortion rate is small but the success rate is large. Regarding the ℓ_2 attacks (e.g., LBFGS, CW), they have a large distortion rate but a satisfactory success rate. Finally, ℓ_∞ attacks (e.g., FGSM) propose to add more disturbance to original images. Hence, their distortion rates are much larger. However, their success rates are not good.

Regarding defensive techniques, BDC can effectively protect gray images with clear boundaries against the eight attacks. Moreover, it is also suitable to eliminate tiny disturbances. For example, we compress a gray-scale image to 1-bit using the truncation value of 0.5. Then, the disturbance with a range of ± 0.2 does not affect the compression value of image pixels in the range of $[0, 0.3)$ and $[0.7, 1)$. However, BDC has limitations for colorful images because excessive bit-depth compressing will make the image recognition a big challenge. Thus, we argue that we should make a tradeoff between maintaining the accuracy of learning models and reducing the success rate of attacks when using BDC. In addition, it is rational that GF and MS are much more effective to defend against ℓ_0 attacks (e.g., JSMA) compared with BDC. The reason is that the goal of ℓ_0 attacks is to add salt-like noise to target images, resulting in a limited number

of changed pixels, and maximizing the change of the value of pixels. Therefore, it is easy for GF and MS to filter such noise. Regarding prediction deviation, we can see from the comparative results that the value deviation of δ_{AD} among different defensive methods is large. Thus, δ_{AD} is a good indicator to distinguish different method from each other when handling adversarial examples in deep learning based image classification. Finally, it is worth noticing that the combined defensive technique is a good choice to defeat multiple types of adversarial attacks in practical usage. Since each defensive approach has its advantages and disadvantages when defending against a variety of attacks, integrating different defensive methods together is meaningful to select a proper method to handle a specific attack.

5. Conclusions

In this paper, we adopt image classification as an application scenario to investigate the security assessment for deep learning with the presence of adversarial examples. We propose EDLIC, a security evaluation framework, which introduces a set of evaluating metrics for comprehensively quantitative comparisons among different adversarial methods and different defensive techniques. Extensive experiments over four well-known image classification datasets, i.e., MNIST, FASHION-MNIST, SVHN, CIFAR-10 and CIFAR-100 have been conducted to demonstrate the effectiveness of EDLIC. By analyzing the comparative results, some interesting findings regarding the selection of deep learning models for image classification are obtained.

In future, we will validate the scalability of the proposed EDLIC framework towards other deep learning-based application scenarios. Moreover, it is meaningful to implement EDLIC in an autonomous security assessment system and then to test its real performance of assessing the security of deep learning based image classifiers in different adversarial environments.

Acknowledgments

The work is supported by National Natural Science Foundation of China under Grant Nos. 61702539 and U1811462, Hunan Provincial Natural Science Foundation of China under Grant No. 2018JJ3611, Shenzhen Fundamental Research Fund under grants No. KQTD2015033114415450 and No. ZDSYS201707251409055, NUDT Research Project under Grant No. ZK-18-03-47, and by grant No. 2017ZT07X152, National Key Research and Development Program of China under Grant No. 2018YFB0204301, National Natural Science Foundation of China under Grant No. 61601483, the Training Program for Excellent Young Innovators of Changsha under Grant No. kq1905006.

References

[1] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural networks, vol.61, pp.85–117, 2015.

[2] Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu, and V.C.M. Leung, "A survey on security threats and defensive techniques of machine learning: A Data Driven View," IEEE Access, vol.6, pp.12103–12117, 2018.

[3] G. Litjens, T, Kooi, B.E, Bejnordi, A.A.A. Setio, F. Ciompi, M. Ghafoorian, J.A.W.M.V. Laak, B. van Ginneken, and C.I. Sánchez, "A survey on deep learning in medical image analysis," Pattern Recognition, vol.42, pp.60–88, 2017.

[4] G.L. Wittel and S.F. Wu, "On attacking statistical spam filters," Proc. CEAS, 2004.

[5] B. Biggio and F. Roli, "Wild patterns: ten years after the rise of adversarial machine learning," Pattern Recognition, vol.84, pp.317–331, 2018.

[6] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," [Online]. Available: https://arxiv.org/abs/1804.08598.

[7] M. Barreno, B. Nelson, R. Sears, A.D. Jpseph, and J.D. Tygar, "Can machine learning be secure?," Proc. ACM Symp. Inf. Comput. Commun. Security (ASIACCS'16), pp.16–25, 2016.

[8] T.W. Weng, H, Zhang, P.Y. Chen, et al, "Evaluating the robustness of neural networks: An extreme value theory approach," Proc. International Conference on Learning Representations (ICLR'18), 2018.

[9] O. Bastani, Y. Ioannou, L. Lampropoulos, D. Vytiniotis, A.V. Nori, and A. Criminist, "Measuring neural net robustness with constraints," Proc. Neural Information Processing Systems (NISP'16), pp.2613–2621, 2016.

[10] V. Tjeng, K. Xiao, and R. Tedrake, "Evaluating robustness of neural networks with mixed integer programming," [Online]. Available: https://arxiv.org/abs/1711.07356.

[11] M, Cheng, J Yi, H, Zhang, et al, "Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples," [Online]. Available: https://arxiv.org/abs/1803.01128.

[12] I.J. Goodfellow, J. Shlens and C. Szegedy, "Explaining and harnessing adversarial examples," Proc. International Conference on Learning Representations (ICLR'15), 2015.

[13] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," Proc. IEEE Symposium on Security and Privacy (SP'17), Heidelberg, pp.39–57, 2017.

[14] S.-M. Moosavi-Dezfooli, A, Fawzi, P, Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," Proc. Computer Vision and Pattern Recognition (CVPR'16), pp.2574–2582, 2016.

[15] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z.B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," Proc. Eur. Symp. Security Privacy (EuroS&P'16), pp.372–387, 2016.

[16] A. Kurakin and I.J. Goodfellow, "Adversarial examples in physical world," [Online]. Available: https://arxiv.org/abs/1607.02533.

[17] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR'2018), pp.9185–9193, 2018.

[18] A. Madry and A. Makelov, "Towards deep learning models resistant to adversaria," [Online]. Available: https://arxiv.org/abs/1706.06083.

[19] P. Chen and Y. Sharma, "EAD: Elastic-net attacks to deep neural networks via adversarial examples," Proc. The Association for the Advancement of Artificial Intelligence (AAAI'18), 2018.

[20] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," Proc. International Conference on Machine Learning (ICML'18), pp.274–283, 2018.

[21] S. Chen, M. Xue, L. Fan, S. Hao, L. Xu, H. Zhu, and B. Li, "Automated Poisoning Attacks and Defenses in Malware Detection Systems: An Adversarial Machine Learning Approach," computers & security, vol.73, pp.326–344, 2018.

[22] C. Liu, B. Li, Y. Vorobeychik, and A. Oprea, "Robust linear regression against training data poisoning," Proc. Artificial Intelligence and Security (AIsec'17), Boston, MA, USA, pp.91–102, 2017.

[23] B. Biggio, I. Corona, G. Fumera, G. Giacinto, and F. Roli, "Bagging classifiers for fighting poisoning attacks in adversarial classification tasks," Proc. Multiple Classifier Systems (MCS'11), Springer,

Berlin, Heidelberg, pp.350–359, 2011.
[24] F. Tramer and A. Kurakin, "Ensemble adversarial training: Attacks and defenses," Proc. International Conference on Learning Representations (ICLR'2018), 2018.
[25] N. Das, M. Shanbhogue, S.-T. Chen, F. Hohman, S. Li, L. Chen, M.E. Kounavis, and D.H. Chau, "Shield: Fast, practical defense and vaccination for deep learning using jpeg compression," Proc. International Conference on Knowledge Discovery & Data Mining (SIGKDD'18), pp.196–204, 2018.
[26] B., Battista, G. Fumera, and F. Roli. "Security evaluation of pattern classifiers under attack," IEEE Transactions on Knowledge and Data Engineering, vol.26, no.4, pp.984–996, 2013.
[27] T.-W. Weng, H. Zhang, P.-Y. Chen, J. Yi, D. Su, Y. Gao, C.-J. Hsieh, and L. Daniel, "Evaluating the robustness of neural networks: An extreme value theory approach," arXiv preprint arXiv:1801.10578, 2018.

Ye Peng received the B.Eng. degree from the Huazhong University of Science and Technology in 2017. She is currently pursuing the M.S. degree with the National University of Defense Technology. Her research interests include cybersecurity and network troubleshooting.

Wentao Zhao received his Ph.D. degree from National University of Defense Technology (NUDT) in 2009. He is now a Professor at NUDT. His research interests include network performance optimization, information processing and machine learning. Since 2011, Dr. Zhao has been serving as a member of Council Committee of Postgraduate Entrance Examination of Computer Science and Technology, NUDT. He has edited one book entitled "Database Principle and Technology" and several technical papers such as KSII TIIS, Communications of the CCF, IJCNN'19, ICC'18, WCNC'17, ICANN'17, WF-IoT, MDAI, FAW.

Wei Cai received the B.Eng. degree in Software Engineering from Xiamen University, China in 2008, the M.S. degree in Electrical Engineering and Computer science from Seoul National University, Korea, in 2011, and the Ph.D. degree in Electrical and Computer Engineering from The University of British Columbia (UBC), Vancouver, Canada, in 2016. From 2016 to 2018, he was a Postdoctoral Research Fellow with UBC. He joined the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, in August 2018, where he is currently an Assistant Professor. He has completed visiting research at National Institute of Informatics, Japan, The Hong Kong Polytechnic University, and Academia Sinica, Taiwan. His recent research interests include software systems, cloud and edge systems, blockchain systems, and game systems. Dr. Cai was a recipient of the 2015 Chinese Government Award for the Outstanding Self-Financed Students Abroad, UBC Doctoral Four-Year-Fellowship from 2011 to 2015, and the Brain Korea 21 Scholarship. He also received the best paper awards from CloudCom2014, SmartComp2014, and CloudComp2013.

Jinshu Su received the B.S. degree in mathematics from Nankai University, Tianjin, China, in 1985, and the M.S. and Ph.D. degrees in computer science from the National University of Defense Technology, Changsha, China, in 1988 and 2000, respectively. He is a Professor with the School of Computer Science, National University of Defense Technology. He currently leads the Distributed Computing and High-Performance Router Laboratory and the Computer Networks and Information Security Laboratory. He also leads the High-Performance Computer Networks Laboratory, which is a key laboratory of Hunan Province, China. His research interests include Internet architecture and network security.

Biao Han is an Associate Professor of Computer Science at National University of Defense Technology (NUDT), China. He received the Ph.D. degree in Computer Science from the University of Tsukuba (Japan) in 2013, the MS and BS degrees from NUDT in 2007 and 2009, respectively. His research mainly focus on cybersecurity for IoT and computer networks, including big data privacy, AI-based networking and wireless communications. He has published over 50 peer reviewed papers in top journals such as JSAC, TPDS, ComNets, TVT and top conferences such as INFOCOM and SIGCOMM. He is a recipient of the Best Paper Award at IEEE LANMAN'2014. He is a member of IEEE and ACM.

Qiang Liu received his Ph.D. degree in computer science and technology from National University of Defense Technology (NUDT) in 2014. From Sep. 2011 to Sep. 2013, he was a Visiting Scholar in Department of Electrical and Computer Engineering at the University of British Columbia (UBC), Canada. He is now an Assistant Professor at NUDT and a member of IEEE and China Computer Federation (CCF). He is also a member of CCF Technical Committee of Network and Data Communications (CCF TCCOMM) and CCF Technical Committee of Theoretical Computer Science (CCF TCTCS). His research interests include 5G network, Internet of Things, wireless network security, and machine learning. Dr. Liu has contributed several archived journals and international conference papers, such as IEEE Access, IEEE Netw., IEEE Trans. Wireless Commun., IEEE Trans. Cybern., IEEE Trans. Knowl. Data Eng., Pattern Recogn., IEEE Commun. Lett., Neurocomputing, AAAI'19, ACM MM'18, ICC'18, EDBT'17, WCNC'17, etc. He currently serves on the editorial review board of Artificial Intelligence Research journal. He had served as a guest editor of some international journals (e.g., MONET, WINET, Mobile Information Systems, etc.), and a Co-Chair of 5GWN'19, ICAIS'19 and ICCCS'18 Workshops.