Adversarial Examples: Opportunities and Challenges

Jiliang Zhang, *Member, IEEE*, Chen Li

Abstract—With the advent of the era of artificial intelligence (AI), deep neural networks (DNNs) have shown huge superiority over human in image recognition, speech processing, autonomous vehicles and medical diagnosis. However, recent studies indicate that DNNs are vulnerable to adversarial examples (AEs) which are designed by attackers to fool deep learning models. Different from real examples, AEs can hardly be distinguished by human eyes, but mislead the model to predict incorrect outputs and therefore threaten security critical deep-learning applications. In recent years, the generation and defense of AEs have become a research hotspot in the field of AI security. This article reviews the latest research progress of AEs. First, we introduce the concept, cause, characteristics and evaluation metrics of AEs, then give a survey on the state-of-the-art AE generation methods with the discussion of advantages and disadvantages. After that, we review the existing defenses and discuss their limitations. Finally, the future research opportunities and challenges on AEs are prospected.

Index Terms—Artificial intelligence (AI), Deep neural networks (DNNs), Adversarial examples (AEs).

I. INTRODUCTION

IN the era of artificial intelligence (AI), deep neural networks (DNNs) have shown great advantages in autonomous vehicles, robotics, network security, image/speech recognition and natural language processing. For example, in 2017, an intelligent robot with the superior face recognition ability, named XiaoDu developed by Baidu, defeated a representative from the team of humans strongest brain with the score of 3:2 [1]. On October 19th, 2017, AlphaGo Zero released by the DeepMind team of Google shocked the world. Compared with the previous AlphaGo, AlphaGo Zero relies on reinforcement learning without any priori knowledge to grow chess skills and finally beats every human competitor [2].

For AI research, the United States received huge support from the government, such as the Federal Research Fund. In October 2016, the United States issued the projects of *Preparing for the Future of Artificial Intelligence* and the *National Artificial Intelligence Research and Development Strategic Plan*, which raised AI to the national strategic level and formulated ambitious blueprints [3], [4]. In 2017, China issued the *New Generation Artificial Intelligence Development Plan*, which mentioned that the scale of the AI core industries would exceed 150 billion CNY by 2020, promoting the development of related industries to enlarge their scale more

J. Zhang and C. Li are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: zhangjiliang@hnu.edu.cn).
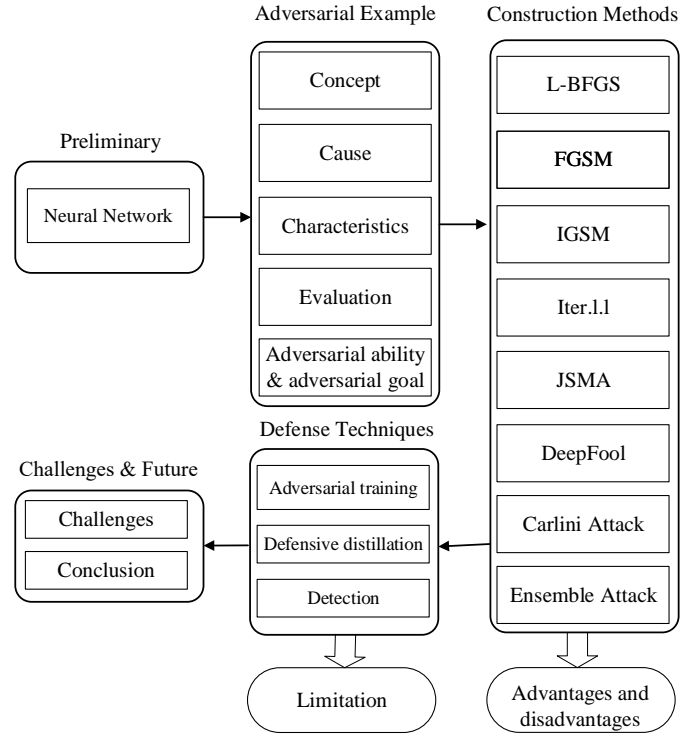

Fig. 1. The general framework towards adversarial examples for this review.

than 1 trillion CNY. In the same year, AI was written into the nineteenth National Congress report, which pushed the development of AI industries to a new height and filled the gap in the top-level strategy of AI development [5].

In the early stage of AI, people pay more attention to the basic theory and application research. With the rapid development of AI, security issues have attracted great attention. For example, at the Shenzhen Hi-tech Fair on November 16, 2016, a robot named Chubby suddenly broke down, hitting the booth glass and hurting pedestrians without any instruction, which is the first robotic injury incident in the world [6]. Soon after, a crime-killing robot, Knightscope, manufactured by Silicon Valley Robotics, knocked down and injured a 16-month-old boy at the Silicon Valley shopping center [7]. At 22 o'clock, March 22, 2018, an Uber autonomous test vehicle hit the 49-year-old woman named Elaine Herzberg who was died after being sent to the hospital for invalid treatment in the suburbs of Tempe, Arizona. This is the first accidental autonomous vehicles accident in the world [8].

In the past decade, various attacks on AI systems have emerged [9]–[16]. At the training stage, poisoning attacks [11]–[13] can damage the original probability distribution of training data by injecting malicious examples to reduce the prediction accuracy of model. At the test or inference
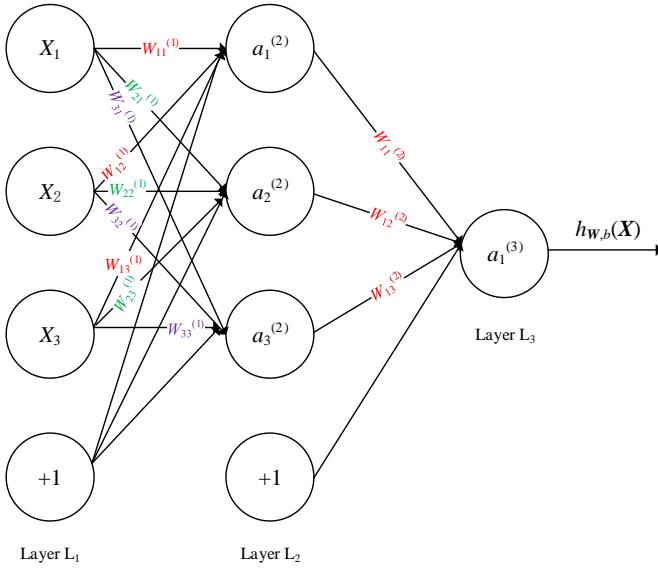
Fig. 2. A three-layer neural network model with the known inputs and weights.

Fig. 3. Generating an adversarial example with the fast gradient sign method, .007 corresponds to a small value ε that restricts the norm of the perturbation, $sign(\nabla_X J(\boldsymbol{\theta}, \boldsymbol{X}, y))$ represents an imperceptibly perturbation [25].

stage, evasion attacks [14]–[16] can trick a target system by constructing a specific input example without changing the target machine learning system. In 2005, Lowd and Meek [17] proposed the concept of adversarial learning in which an adversary conducts an attack that minimizes a cost function. Under the framework, they presented algorithms for an attacker to reverse engineer linear classifiers. In 2006, Barreno et al. [18] presented a taxonomy of different types of attacks on machine learning systems. In order to mitigate poisoning attacks and evasion attacks, a lot of defenses have been proposed [9], [10], [19]–[21]. In 2014, Szegedy et al. [22] proposed the concept of *adversarial example* (AE), which adds a slight perturbation to the input, resulting in the adversarial image being misclassified by the model with high confidence, while the human eyes cannot recognize the difference. Even though different models have different architectures and training data, the same set of AEs can be used to attack all related models. AEs have shown a huge threat to DNNs. For example, the classifier may misclassify an adversarial image of the stop traffic sign as a speed limit sign of 45km/h, resulting in a serious traffic accident [23]. In image captioning system, an image is used as input to generate some captions to describe the image which is perturbed by attackers to generate some image-independent, completely opposite or even malicious captions [24].

In recent years, many AE construction methods and defense techniques have been proposed. This survey elaborates on the related research and development status of AE on DNNs since it was proposed in [22]. The overall framework is shown in Fig. 1.

II. PRELIMINARIES

A. Neural Network

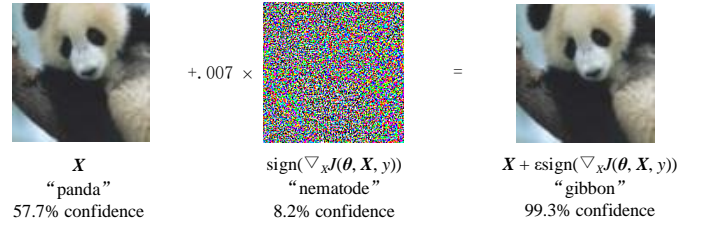Artificial neural network (ANN) is a kind of operational model that simulates the human brain's nervous system to process complex information, consisting of many neurons connected to each other. Each neuron represents a specific output function called the activation function. The connection between two neurons represents the weight value of the signal. The neural network connects many single neurons together by weights to simulate the human brain to process information.

A simple neural network consisting of an input layer L_1, a hidden layer L_2 and an output layer L_3 is shown in Fig. 2, where the circle represents the neuron of the neural network; the circle labelled "+1" represents the bias unit; the circles labelled "X_1", "X_2", "X_3" are the inputs. Neurons in different layers are connected by weights $\boldsymbol{W}$. We use $a_i^{(l)}$ to represent the activation value (output value) of the *i-th* unit in *l-th* layer, when $l = 1$, $a_i^{(1)} = X_i$. With the given inputs and weights, the function output $h_{(\boldsymbol{W},b)}(\boldsymbol{X})$ can be calculated. The specific steps are as follows:

$$a_1^{(2)} = f(W_{11}^{(1)}X_1 + W_{12}^{(1)}X_2 + W_{13}^{(1)}X_3 + b_{11}^{(1)}) \quad (1)$$

$$a_2^{(2)} = f(W_{21}^{(1)}X_1 + W_{22}^{(1)}X_2 + W_{23}^{(1)}X_3 + b_{21}^{(1)}) \quad (2)$$

$$a_3^{(2)} = f(W_{31}^{(1)}X_1 + W_{32}^{(1)}X_2 + W_{33}^{(1)}X_3 + b_{31}^{(1)}) \quad (3)$$

$$a_1^{(3)} = h_{\boldsymbol{W},b}(\boldsymbol{X}) = f(W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_3^{(2)} + b_{11}^{(2)}) \quad (4)$$

The above calculation process is called forward propagation (FP), which is a transfer process of input information through the hidden layer to the output layer. The activation function ReLU: $f(X) = max\{0, X\}$ is used to nonlinearize the neural network between different hidden layers. When the machine learning task is a binary classification, the final output layer uses the activation function sigmoid: $f(X) = 1/(1 + e^{(-X)})$; when the machine learning task is a multi-class problem, the final output layer uses the activation function softmax: $f(X) = \frac{e^{X_k}}{\sum_{i=1}^{N} e^{X_i}}$, $k = 1, 2, ...N$. In the training process, the weights $\boldsymbol{W}$ and the bias b connecting the neurons in different layers are determined by back propagation.

Neural networks belong to a cross-disciplinary research field combining computer, probability, statistics, and brain science. It focuses on how to enable computers to simulate and implement human learning behaviors, thus achieving better automatic knowledge acquisition. However, recent studies show that neural networks are particularly vulnerable to AEs which are generated by adding small perturbations to the inputs. In what follows, we will discuss the AEs in detail.
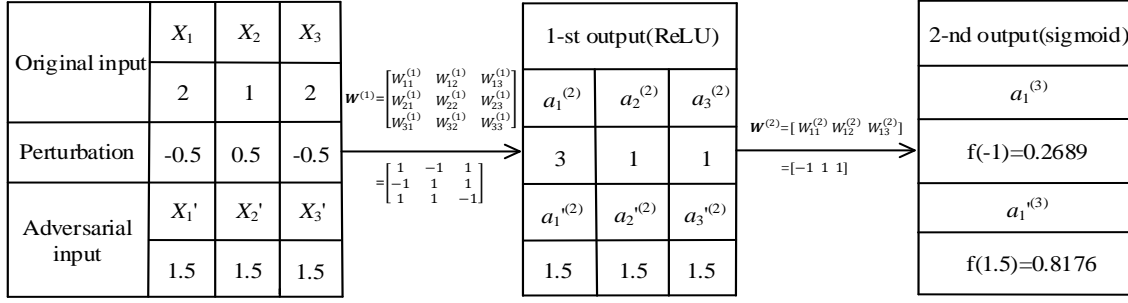
Fig. 4. The perturbation to the original inputs for a three-layer neural network (Fig. 2). The value of the original inputs X_1, X_2, X_3 and the weights $\boldsymbol{W}^{(1)}, \boldsymbol{W}^{(2)}$ are initialized randomly.

III. ADVERSARIAL EXAMPLES

In 2014, Szegedy et al. [22] proposed adversarial examples to fool DNNs. A subtle perturbation is added to the input of the neural network to produce a wrong output with high confidence, while the human eyes cannot recognize the difference. Suppose that there is a machine learning model M and an original example C which can be correctly classified by the model, i.e., $M(C) = y_{true}$, where y_{true} is the true label of C. However, it is possible to construct an adversarial example C' which is perceptually indistinguishable from C but is classified incorrectly, i.e., $M(C') \neq y_{true}$ [22]. A typical example is shown in Fig. 3, the model considers the original image to be a "panda" (57.7%). After adding a slight perturbation to the original image, the original image is classified as a "gibbon" by the same model with 99.3% confidence, while the human eyes completely cannot distinguish the differences between the original image and the adversarial image [25].

In order to facilitate the reader to understand AEs intuitively, we use the neural network model in Fig. 2 as an example to show the change of the outputs by perturbing the inputs. As shown in Fig. 4, $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$ are the weight matrices. After adding a small perturbation sign (0.5) to the original inputs, the adversarial inputs X'_1, X'_2, X'_3 are equal to 1.5. Then, after going through the first layer's weight matrix $\boldsymbol{W}^{(1)}$ and the activation function ReLU transform operation, the 1-st layer output $a'^{(2)}_1, a'^{(2)}_2, a'^{(2)}_3$ are equal to 1.5. Finally, after passing the second layer's weight matrix $\boldsymbol{W}^{(2)}$ and the activation function sigmoid transform operation, the probability of the output is changed from 0.2689 to 0.8176, which makes the model misclassify the image with high confidence. With the increase of the model depth, the probability of the output changes more obviously.

A. Cause of Adversarial Examples

The AE is a serious vulnerability in deep learning systems and cannot be ignored in security-critical AI applications. However, in the current research, there are no well-recognized explanations why the AEs can be constructed. Analyzing the cause of AEs can effectively help researchers to fix the vulnerability effectively. Some researchers [26] suspected that the reason is over-fitting or insufficient regularization of the model which leads to insufficient generalization ability that learning models predict unknown data. However, by adding

Fig. 5. The probability of class '1' before and after the perturbation.

perturbations to a regularized model, Goodfellow et al. [25] found that the effectiveness against AEs was not be improved significantly. Other researchers [27] suspected that AEs arose from extreme nonlinearity of deep neural networks. However, if the inputs of a linear model had enough dimensions, AEs can also be constructed successfully with high confidence by adding small perturbations to the inputs [25].

Goodfellow et al. [25] believed that the reason for generating AEs is the linear behavior in high dimensional space. In the high dimensional linear classifier, each individual input feature is normalized. For one dimension of each input, small perturbations will not change the overall prediction of the classifier. However, small perturbations to all dimensions of the inputs will lead to an effective change of the output.

As shown in Fig. 5, in a particular direction, by adding or subtracting 0.5 to each dimension of the original example X, the score of class '1' is improved from 5% to 88%. It demonstrates that linear models are vulnerable to AEs and refutes the hypothesis that the existence of AEs is due to the high nonlinearization of model. Therefore, the existence of the high-dimensional linear space may be the cause of AEs.

B. Characteristics of Adversarial Examples

In general, AEs have three characteristics:

Transferability. AEs are not limited to attack a specific neural network. It is unnecessary to obtain architecture and parameters of the model when constructing AEs, as long as the model is trained to perform the same task. AEs generated from one model M_1 can fool a different model M_2 with a similar probability. Therefore, an attacker can use AEs to attack the models that perform the same task, which means that
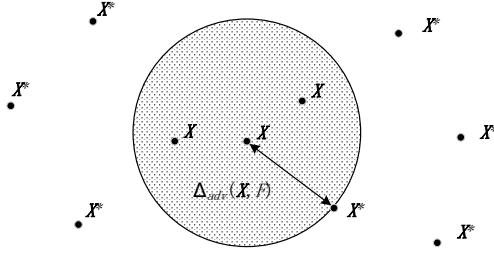
Fig. 6. **Visualizing the metric of robustness:** This 2D representation illustrates the metric as the radius of the disc at the original example X and going through the closest adversarial example X^* constructed from X [30].

an attacker can construct AEs in the known machine learning model and then attack related unknown models [28].

Regularization effect. Adversarial training [25] can reveal the defects of models and improve the robustness of examples. However, compared to other regularization methods, the cost of constructing a large number of AEs for adversarial training is expensive. Unless researchers can find shortcuts for constructing AEs in the future, they are more likely to use dropout [29] or weight decay (L_2 regularization).

Adversarial instability. In the physical world, it is easy to lose its adversarial for AEs after physical transformations such as translation, rotation, and lighting. In this case, AEs will be correctly classified by the model. This instability characteristic challenges attackers to construct robust AEs and creates the difficulty of deploying AEs in the real world.

C. Evaluation Metrics

1) Success Rate: When constructing AEs, the success rate is the most direct and effective evaluation criterion. In general, the success rate to generate AEs is inversely proportional to the magnitude of perturbations. For example, the fast gradient sign method [25] requires a large perturbation and is prone to label leaking [31] that the model correctly classifies the AE generated with true label and misclassifies the AE created with false label. Therefore, the success rate is much lower than the iterative method [32] with the lower perturbation and the Jacobian-based saliency map attack method [30] with the specific perturbation. Usually, it is difficult to construct AEs with 100% success rate.

2) Robustness: The robustness of machine learning models is related to the classification accuracy [33], [34]. Better machine learning models are less vulnerable to AEs. Robustness is a metric to evaluate the resilience of DNNs to AEs. In general, a robust DNN model has two features [35] [36]:

- The model has high accuracy both inside and outside of the dataset;
- The classifier of a smoothing model can classify inputs consistently near a given example.

We first define the robustness of classifiers $f(X)$ to adversarial perturbations in the input space $\mathbb{R}^d$. Given an input $X \in \mathbb{R}^d$ from μ, $\Delta_{adv}(X, F)$ is denoted as the norm of the smallest perturbation to make classifiers misclassified.

$$\Delta_{adv}(X, F) = \min_{r \in \mathbb{R}^d} \|r\|_2$$
$$subject\ to\ f(X)f(X + r) \leq 0 \tag{5}$$

where the perturbation r aims to flip the label of X, corresponding to the minimal distance from X to the decision boundary of the classifier.

The robustness of a DNN model F to adversarial perturbations is defined as the average of $\Delta_{adv}(X, F)$ over all X:

$$\rho_{adv}(F) = E_\mu[\Delta_{adv}(X, F)] \tag{6}$$

As shown in Fig. 6, in the circle with a radius of $\Delta_{adv}(X, F)$, the output of the classifier is a constant original example X, and the output becomes an AE X^* outside the circle. Therefore, the magnitude of the perturbation $\Delta_{adv}(X, F)$ is proportional to the robustness of the model, i.e., the higher the minimum perturbation needed to misclassify the example, the stronger the robustness of the DNN is.

3) Transferability: AEs generated for one machine learning model can be used to misclassify another model even if both models have different architectures and training data. This property is called transferability. AEs can be transferred among different models because a contiguous subspace with a large dimension in the adversarial space is shared among different models [37]. Transferability that attackers only need to train an alternative model to construct AEs and deploy the AEs to attack the target model provides a tremendous advantage for AEs.

The transferability of AEs can be measured by the transfer rate, i.e., the ratio of the number of transferred AEs to the total number of AEs constructed by the original model. In the non-targeted attack, the percentage of the number of AEs generated by one model that are correctly classified by another model is used to measure the non-targeted transferability. It is called as accuracy rate. A lower accuracy rate means better non-targeted transfer rate. In the targeted attack, the percentage of the number of AEs generated for one model that are classified as the target label by the other model is used to measure the targeted transferability. It is referred to as matching rate. A higher matching rate means better targeted transfer rate [38].

The transfer rate of AEs depends on two factors. One factor is the model-related parameters, including model architecture, model capacity, and test accuracy. The transfer rate of AEs is high among models with similar architecture, low model capacity (the number of model parameters) and high test accuracy [39]. Another factor is the magnitude of adversarial perturbation. Within a certain perturbation range, the transfer rate of AEs is proportional to the magnitude of adversarial perturbations, i.e., the greater perturbations to the original example, the higher transfer rate of the constructed AE. The minimum perturbation required for different methods of constructing AEs is different.

4) Perturbations: Too small perturbations to the original examples are difficult to construct AEs while too large perturbations can be distinguished by human eyes easily. Therefore, perturbations need to achieve a balance between constructing AEs and human visual system. For example, it is difficult to control the perturbation for FGSM [25] which incurs label leaking easily. To address the issue, Kurakin et al. proposed an optimized FGSM based on iterative method [32] which can control the perturbation within a threshold range. Hence, the success rate of constructing AEs is improved significantly.
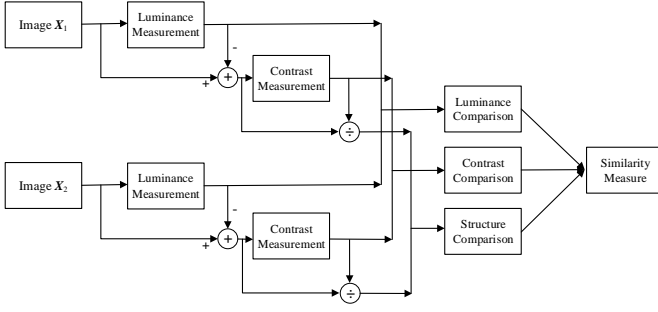
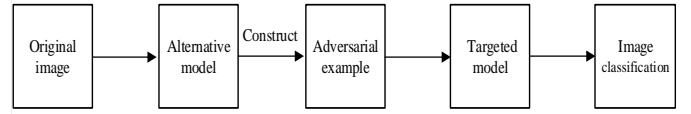Fig. 7. The structural similarity (SSIM) measurement system [40].

Fig. 8. **Black-box attack.** Alternative model: attackers know the model architecture and parameters; targeted model: attackers do not know the model architecture and parameters.

However, the transfer rate of such AEs is low. Later on, a saliency map-based method [30] is proposed to improve the transfer rate. The key steps include: 1) *direction sensitivity estimation*: evaluate the sensitivity of each class for each input feature; 2) *perturbation selection*: use the sensitivity information to select a minimum perturbation δ among the input dimension which is most likely to misclassify the model.

In general, L_2 norm is used to measure the perturbation of the AE:

$$d(\boldsymbol{X}^{'}, \boldsymbol{X}) = ||\boldsymbol{X}^{'} - \boldsymbol{X}||_2 = \sqrt{\sum_{i=1}^{n}(X_i^{'} - X_i)^2} \qquad (7)$$

where the n-dimensional vectors $\boldsymbol{X}$ and $\boldsymbol{X}^{'}$ represent the original example and AE respectively, $d(\boldsymbol{X}^{'}, \boldsymbol{X})$ is the distance metric between $\boldsymbol{X}$ and $\boldsymbol{X}^{'}$. It shows that the larger $d(\boldsymbol{X}^{'}, \boldsymbol{X})$ to the original example, the greater perturbations needed to construct AEs.

5) Perceptual adversarial similarity score (PASS): AEs are not only misclassified by the models, but also visually recognized as correct classes by humans. Since the human visual system is sensitive to structural changes, Wang et al. [40] proposed the structural similarity (SSIM) index as a metric to measure the similarity between two images. Luminance and contrast associated with the object structure are defined as the structure information of the image.

The structure of the SSIM measurement system is shown in Fig. 7. For two different aligned images $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, the SSIM measurement system consists of three comparisons: luminance, contrast, and structure. First, the luminance of each image is compared. Second, the standard deviation (the square root of variance) is used as an estimate of the contrast of each image. Third, the image is normalized by its own standard deviation as an estimate of the structure comparison. Finally, the three components are combined to produce an overall similarity measure. Therefore, the structural similarity between image signal $\boldsymbol{X}$ and image signal $\boldsymbol{Y}$ can be modeled as:

$$SSIM(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{m}\sum_{n=1}^{m}[L(X_n, Y_n)^{\alpha}C(X_n, Y_n)^{\beta}S(X_n, Y_n)^{\gamma}] \qquad (8)$$

where m is the number of pixels; L, C, and S are the luminance, contrast, and structure of the image, respectively; hyper-parameters α, β and γ are used to weight the relative importance of L, C, and S, respectively; the default setting is $\alpha = \beta = \gamma = 1$.

Based on SSIM measurement system, Perceptual Adversarial Similarity Score (PASS) is proposed to quantify human perception of AEs [41]. The PASS between $\boldsymbol{X}$ and $\boldsymbol{X}^{'}$ is defined as:

$$PASS(\boldsymbol{X}^{'}, \boldsymbol{X}) = SSIM(\psi(\boldsymbol{X}^{'}, \boldsymbol{X}), \boldsymbol{X}) \qquad (9)$$

where $\psi(\boldsymbol{X}^{'}, \boldsymbol{X})$ represents the homography transform (a mapping from one plane to another) from the original image $\boldsymbol{X}$ to the adversarial image $\boldsymbol{X}^{'}$. PASS can quantify the AEs by measuring the similarity of the original image and the adversarial image. An appropriate PASS threshold can be set to distinguish the AEs with excessive perturbations. Meanwhile, attackers can also use the PASS threshold to optimize the methods of constructing AEs. Therefore, constructing an AE should satisfy:

$$\underset{d(\boldsymbol{X}, \boldsymbol{X}^{'})}{argmin} \ \boldsymbol{X}^{'} : f(\boldsymbol{X}^{'}) \neq y \ and \ PASS(\boldsymbol{X}, \boldsymbol{X}^{'}) \geq \theta \qquad (10)$$

where $d(\boldsymbol{X}, \boldsymbol{X}^{'})$ is some dissimilarity measure, $f(\boldsymbol{X}^{'}) \neq y$ represents that the AE is misclassified by the model, θ is the PASS threshold set by the attacker, $PASS(\boldsymbol{X}, \boldsymbol{X}^{'}) \geq \theta$ represents that the AE is not recognized by the human eyes.

D. Adversarial Abilities and Adversarial Goals

Adversarial ability is determined by how well attackers understand the model. Threat models in deep learning system are classified into the following types according to the attacker's abilities.

White-box attack. Attackers knows everything related to trained neural network models, including training data, model parameters and model architectures.

Grey-box attack. Attackers know some model information such as model architectures, learning rate, training data and training steps, except model parameters. This attack is a byproduct of black-box attack and is still not common in practical applications.

Black-box attack. Attackers do not know the architecture and parameters of the machine learning model, but can interact with the machine learning system. For example, the output can be classified by randomly test vectors. Attackers utilize the transferability of AE to train an alternative model to construct AEs first, and then use the generated AEs to attack the unknown target model.

The process of black-box attack is shown in Fig. 8. First, attackers use the known dataset to train an alternative model. Then, attackers construct the corresponding AE through the alternative model. Finally, based on the transferability, attackers

construct AEs with known machine learning model to attack the unknown target model. However, in some scenarios such as machine learning as a service, it is difficult to obtain the structure and parameters of target model and training dataset. Papernot et al. [28] proposed a practical black-box attack method to generate AEs. The specific process is as follows.

1) Attackers use the target model as an oracle to construct a synthetic dataset, where the inputs are synthetically generated and the outputs are labels observed from the oracle.
2) Attackers select a machine learning algorithm randomly and use the synthetic dataset to train a substitute model.
3) Attackers generate AEs through the substitute model.
4) Based on the transferability, attackers use the generated AEs to attack the unknown target model.

In adversarial deep learning, the adversarial goal is to make the model misclassify the output. According to the different influence of the perturbation on the classifier, we classify the adversarial goals into four types:

(1) **Confidence reduction:** reduce the confidence of output classification.
(2) **Non-targeted misclassification:** alter the output classification to any class which is different from the original class.
(3) **Targeted misclassification:** force the output classification to be the attacker-specific target class.
(4) **Source/Target misclassification:** select a specific input to generate an attacker-specific target class.

As shown in Fig. 9, the vertical axis represents the adversarial abilities which include architecture, training data, oracle (according to the relationship of change between the inputs and the outputs provided by the model to generate AEs), and examples (just know the inputs and the outputs but cannot modify these inputs to observe the difference in the outputs). The horizontal axis represents the adversarial goals, and the increasing complexity from left to right is confidence reduction, non-targeted misclassification, targeted misclassification, and source/target misclassification. In general, the weaker the adversarial ability or the higher the adversarial goal, the more difficult the model to be attacked.

IV. METHODS OF CONSTRUCTING AEs

In what follows, several typical AE methods will be introduced in detail.

A. Mainstream Attack Methods

1) L-BFGS

Szegedy et al. [22] proposed L-BFGS (Limited-memory; Broy-den, Fletcher, Goldforb, Shanno) to construct AEs. Given an image X, attackers construct an image X' similar to X with L_2 norm and X' can be labelled as different class. The optimization problem is:

$$minimize \quad ||X - X'||_2^2 \tag{11}$$

where $||X - X'||_2^2$ is L_2 norm. The attack goal is to make $f(X') = l, X' \in [0,1]^n$, where l is the target class. $f(X') = l$
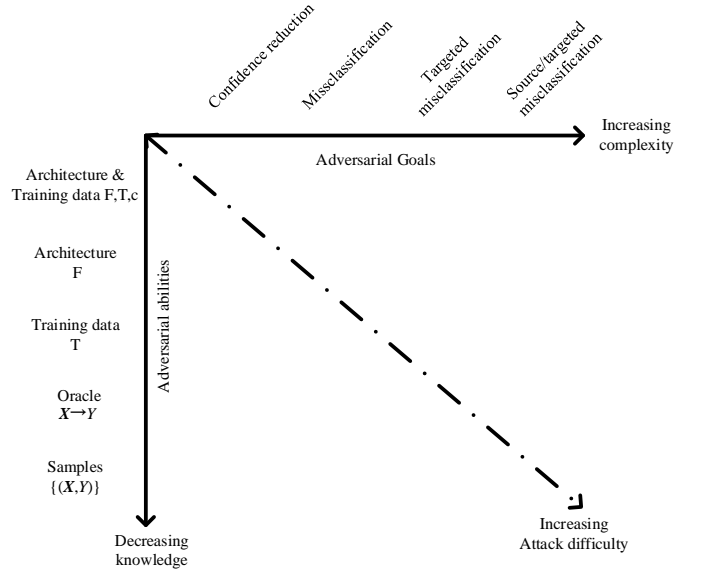

Fig. 9. Adversarial abilities and adversarial goals [30].

is the nonlinear and non-convex function which is difficult to be solved directly. Therefore, the problem is solved approximately by the box-constrained L-BFGS.

$$minimize \quad c \cdot ||X - X'||_2^2 + loss_{F,l}(X') \tag{12}$$

where c is a randomly initialized hyper-parameter, which is determined by linear search; $loss_{F,l}(*)$ is the loss function. $f(X') = l$ is approximated by minimizing the loss function. Although this method is high stability and effectiveness, the calculation is complicated.

2) FGSM

Goodfellow et al. [25] proposed a simplest and fastest method to construct AEs, named fast gradient sign method (FGSM). The generated images are misclassified by adding perturbations and linearizing the cost function in the gradient direction. Given an original image X, the problem can be solved with:

$$X^{adv} = X + \varepsilon sign(\nabla_X J(X, y_{true})) \tag{13}$$

where X^{adv} represents an AE from X, ε is a randomly initialized hyper-parameter, $sign(*)$ is a sign function, y_{true} is the true label corresponding to X, and $J(*)$ is the cost function used to train the neural network, $\nabla X(*)$ represents the gradient of X.

There are two main differences between FGSM and L-BFGS. First, FGSM is optimized with the L_∞ norm. Second, FGSM is a fast AE-construction method because it does not require an iterative procedure to compute AEs. Hence it has lower computation cost than other methods. However, FGSM is prone to label leaking. Therefore, Kurakin et al. [31] proposed FGSM-pred which uses the predicted label y_{pred} instead of true label y_{true}. Researchers [31] also use the gradients with L_2 and L_∞ norm, i.e., $sign(\nabla_X J(X, y_{true}))$ is changed to $\frac{sign(\nabla_X J(X, y_{true}))}{||sign(\nabla_X J(X, y_{true}))||_2}$ and $\frac{sign(\nabla_X J(X, y_{true}))}{||sign(\nabla_X J(X, y_{true}))||_\infty}$, these two methods are named as FGSM grad.L_2 and FGSM grad.L_∞, respectively.
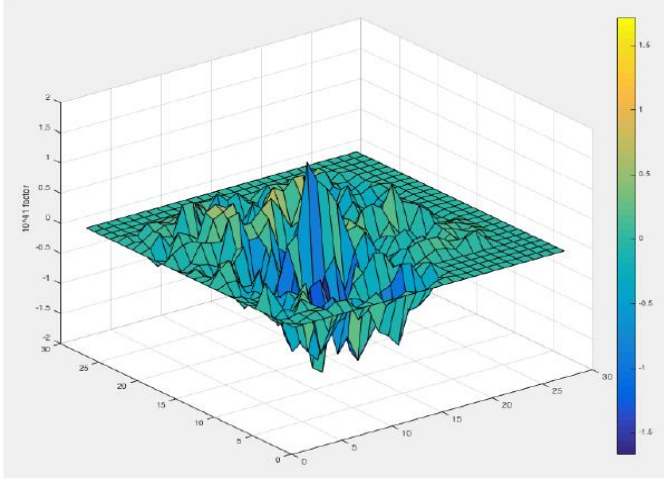
Fig. 10. Saliency map S with the 28×28 image pixel, large absolute values correspond to features with a significant impact on the output when perturbed the input [30].

3) *IGSM*

It is difficult for FGSM to control the perturbation in constructing AEs. Kurakin et al. [32] proposed an optimized FGSM, named iterative gradient sign method (IGSM), which applies perturbations to multiple smaller steps and clips the results after each iteration to ensure that the perturbations are within the neighborhood of the original image. For the *N-th* iteration, the update process is:

$$X_0^{adv} = X, X_{N+1}^{adv} = Clip_{X,\epsilon}\{X_N^{adv} + \varepsilon sign(\nabla_X J(X_N^{adv}, y_{true}))\} \quad (14)$$

where $Clip_{X,\epsilon}(*)$ denotes $[X - \epsilon, X + \epsilon]$.

IGSM is non-linear in the gradient direction and requires multiple iterations, which is simpler than L-BFGS method in calculation, and the success rate of AE construction is higher than FGSM. IGSM can be further divided into two types: 1) reducing the confidence of the original prediction as the original class; 2) increasing the confidence of the prediction that originally belongs to the class with smallest probability. The iterative least-likely class method that will be introduced in the next section is the second case.

Recently, Dong et al. [47] proposed a momentum iterative method (MIM). The basic idea is to add momentum based on the IGSM. The weakness of previous iterative attacks is that the transferability (black-box attack) is weakened when the number of iterations increases, which can be addressed after adding momentum in iterative attacks. MIM attack not only enhances the attack ability on the white-box model, but also increases the success rate on black-box model. The momentum iterative gradient sign method for targeted attack is given by

$$g_{t+1} = \mu g_t + \frac{\nabla_X J(X_t^{adv}, y_{target})}{||\nabla_X J(X_t^{adv}, y_{target})||_1}$$
$$X_{t+1} = Clip_{[0,1]}(X_t + \alpha sign(g_{t+1})) \quad (15)$$

where μ is the decay factor of momentum.

4) *Iterl.l*

FGSM and L-BFGS try to increase the probability of predicting wrong results, but do not specify which wrong class should be selected by the model. These methods are sufficient for small datasets such as MNIST and CIFAR-10. On ImageNet, with a larger number of classes and the varying degrees of significance between classes, FGSM and L-BFGS may construct uninteresting misclassifications, such as misclassifying one type of cat into another cat. To generate more meaningful AEs, a novel AE generation method is proposed by perturbing the target class with the lowest probability so that this least-likely class turns to become the correct class after the perturbation, which is called as iterative least-likely class method (*iterl.l*) [32]. To make the adversarial image X^{adv} be classified as y_{LL}, we have the following procedure:

$$X_0^{adv} = X, X_{N+1}^{adv} = Clip_{X,\epsilon}\{X_N^{adv} - \varepsilon sign(\nabla_X J(X_N^{adv}, y_{LL}))\} \quad (16)$$

where y_{LL} represents the least likely (the lowest probability) target class. For a classifier with good performance, the least likely class is usually quite different from the correct class. Therefore, this attack method can lead to some interesting errors, such as misclassifying a cat as an aircraft. It is also possible to use a random class as the target class, which is called as iteration random class method.

5) *JSMA*

Papernot et al. [30] proposed Jacobian-based Saliency Map Attack (JSMA), which is based on the L_0 distance norm. The basic idea is to construct a saliency map with the gradients and model the gradients based on the impact of each pixel. The gradients are directly proportional to the probability that the image is classified as the target class, i.e., changing a large value will significantly increase the likelihood that the model classifies the image as the target class. JSMA allows us to select the most important pixel (the maximum gradient) based on the saliency map and then perturb the pixel to increase the likelihood of labeling the image as the target class. More specifically, JSMA includes the following steps:

(1) Compute forward derivative $\nabla F(X)$.

$$\nabla F(X) = \frac{dF(X)}{dX} = [\frac{dF_j(X)}{dX_i}]_{i \in 1...M, j \in 1...N} \quad (17)$$

(2) Construct a saliency map S based on the forward derivative, as shown in Fig. 10.

(3) Modify the most important pixel based on the saliency map, repeat this process until the output is the target class or the maximum perturbation is got.

When the model is sensitive to the change of inputs, JSMA is easier to calculate the minimum perturbation to generate the AEs. JSMA has high computational complexity while the generated AEs have high success rate and transfer rate.

6) *DeepFool*

Mohsen et al. [44] proposed a non-targeted attack method based on L_2 norm, called DeepFool. Assuming that the neural network is completely linear, there must be a hyperplane separating one class from another. Based on this assumption,
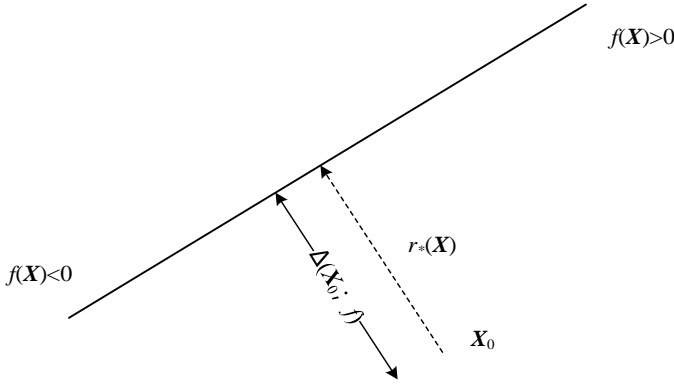
Fig. 11. Adversarial examples for a linear binary classifier [44].

we analyze the optimal solution to this problem and construct AEs. The corresponding optimization problem is as follows:

$$r_*(X_0) = argmin||r||_2 \qquad (18)$$

subject to $sign(f(X_0+r)) \neq sign(f(X_0))$, where r indicates the perturbation.

As shown in Fig. 11, X_0 is the original example, $f(X)$ is a linear binary classifier, the straight line $WX + b = 0$ is the decision boundary, and $r_*(X)$ is the distance from the original example to the decision boundary, i.e., the distance from X_0 to the straight line $WX + b = 0$. The distance is equivalent to the perturbation $\Delta(X_0; f)$. Therefore, when $\Delta(X_0; f) > r_*(X)$, the AE can be generated.

Compared with L-BFGS, DeepFool is more efficient and powerful. The basic idea is to find the decision boundary that is the closest to X in the image space, and then use the boundary to fool the classifier. It is difficult to solve this problem directly in neural networks with high dimension and nonlinear space. Therefore, a linearized approximation is used to iteratively solve this problem. The approximation is to linearize the intermediate X_0 classifier in each iteration, and obtain an optimal update direction on the linearized model. Then X_0 is iteratively updated in this direction by a small step α, repeating the linear update process until X_0 crosses the decision boundary. Finally, AEs can be constructed with subtle perturbations.

7) *Universal Adversarial Perturbations*

FGSM, JSMA and DeepFool can only generate adversarial perturbations to fool a network on a single image. Moosavi-Dezfooli et al. [45] proposed a universal image-agnostic perturbation attack method which fools classifiers by single adversarial perturbation to all images. The specific problem can be defined as finding a universal perturbation v, such that $\hat{k}(X + v) \neq \hat{k}(X)$ for most examples in the dataset subject to the distribution μ.

$$\hat{k}(X + v) \neq \hat{k}(X) \quad for \; ``most" \; X \sim \mu \qquad (19)$$

where $\hat{k}$ denotes a classification function that labels each image. Such perturbation v is called as the universal perturbation. Attackers' goal is to find the v that satisfies the following two constraints:

$$\begin{cases} 1. \; ||v||_p \leq \S \\ 2. \; \underset{X \sim \mu}{\mathbb{P}} \left(\hat{k}(X+v) \neq \hat{k}(X) \right) \geq 1 - \delta \end{cases} \qquad (20)$$

where $|| \cdot ||_p$ denotes the p-norm, the parameter $\S$ controls the magnitude of the perturbation v, and δ quantifies the desired fooling rate for all images. The attack method has two characteristics: 1) the perturbation is related to the target model rather than the image; 2) the small perturbation will not change the structure of the image itself.

8) *CW Attack (C&W)*

Carlini and Wagner [46] proposed a powerful attack method based on L-BFGS. The attack with L_0, L_2, L_∞ distance norm can be targeted or non-targeted, and we take the non-targeted L_2 norm as an example here. The corresponding optimization problem is:

$$minimize \; ||\delta||_2 + c \cdot f(X + \delta) \qquad (21)$$

where $X + \delta \in [0, 1]^n$, c is a hyper-parameter that can balance these two terms, and δ is a small perturbation. The definition of the objective function $f(X^{'})$ is as follows:

$$f(X^{'}) = max(max\{Z(X^{'})_i : i \neq t\} - Z(X^{'})_t, -l) \qquad (22)$$

where $Z(X^{'})$ is the last hidden layer, t is the true label, and l is a hyper-parameter, which is used to control the confidence level of the model misclassification, and the AE $X^{'}$ can be classified as t with high confidence by adjusting the value of l. In general, high confidence attacks have large perturbations and high transfer rates, and CW Attack based on the L_0, L_2, L_∞ distance metric can defeat the defensive distillation [59]. There are three improvements to this attack based on L-BFGS:

- Use the gradient of the actual output in the model instead of the gradient of softmax.
- Apply different distance metrics (L_0, L_2, L_∞).
- Apply different objective functions $f(X^{'})$.

9) *Ensemble Attack*

Liu et al. [38] proposed an ensemble attack method combining multiple models to construct AEs. If an adversarial image remains adversarial for multiple models, it is likely to be transferred to other models. Formally, given k white-box models with softmax outputs being $J_1, ..., J_k$, an original image X and its true label y, the ensemble-based approach solves the following optimization problem (for targeted attack):

$$argmin_{X'} - log((\sum_{i=1}^{k} \alpha_i J_i(X^{'})) \cdot \mathbf{1}_{y'}) + \lambda d(X, X^{'}) \qquad (23)$$

where $y^{'}$ is the target label specified by the attacker, $\sum_{i=1}^{k} \alpha_i J_i(X^{'})$ is the ensemble model, and α_i is the weight of the i-th model, $\sum_{i=1}^{k} \alpha_i = 1$, λ is a randomly initialized parameter that is used to control the weight of the two terms. The goal is to ensure the generated AEs are still adversarial for the other black-box model J_{k+1}. Since the decision boundaries for different models are almost the same, the transferability of targeted AEs is improved significantly.

TABLE I
ADVANTAGES AND DISADVANTAGES OF DIFFERENT ATTACK METHODS

Method	Advantage	Disadvantage
L-BFGS [22]	High stability and effectiveness	High computational and time complexity
FGSM [25]	Low computing complexity, high transfer rate	Low success rate, label leaking
IGSM [32]	Small perturbations, high success rate	Low transfer rate, low success rate in balck-box attacks
Iter.l.l [32]	Small perturbations, high success rate	Low transfer rate, low success rate in balck-box attacks
JSMA [30]	Small perturbations, high success rate	High computational complexity
Uni.perturbations [45]	High generalization ability, high transfer rate	The perturbation is not easy to control, low success rate of target attacks
DeepFool [44]	Low computing complexity, small perturbations	Low success rate in balck-box attacks
One-pixel [50]	Low computing complexity	Low success rate of target attacks, large perturbations
CW Attack [46]	Small perturbations, high transfer rate and success rate	High computational complexity
Ensemble Attack [38]	Simple calculation, good generalization	Low success rate in balck-box attacks

B. Other Attack Methods

From the perspective of attackers, the attack goal is to construct strong AEs with small perturbations and fool the model with high confidence while not being distinguished by human eyes. Recently, in addition to typical AE constructing methods introduced above, a lot of other attack methods have been proposed [80]. Xia et al. [48] proposed AdvGAN to construct AEs. The basic idea is to use generative adversarial networks to construct targeted AEs, which not only learns and preserves the distribution of the original examples, but also guarantees the diversity of perturbations and enhances generalization ability significantly. Tramr et al. [49] proposed an ensemble attack RAND + FGSM. First, they added a small random perturbation RAND to "escape" the non-smooth vicinity of the data point before computing the gradients. Then, they applied the FGSM to enhance the attack ability greatly. Compared with the FGSM, this method has a higher success rate and can effectively avoid label leaking. Su et al. [50] proposed a one pixel attack method which only changes one pixel for each image to construct AEs to fool DNNs. However, such simple perturbation can be recognized by the human eyes easily. Weng et al. [51] proposed a computationally feasible method called Cross Lipschitz Extreme Value for nEtwork Robustness (CLEVER), which applies extreme value theory to estimate a lower bound of the minimum adversarial perturbation required to misclassify the image. CLEVER is the first attack-independent method, and can evaluate the intrinsic robustness of neural networks. Brown et al. [52] proposed adversarial patch which does not need to subtly transform an existing image into another and can be placed anywhere within the field of view of the classifier to cause the classifier to output a targeted class.

As discussed above, many AE attacks have been proposed in recent years. We summarized advantages and disadvantages of several typical attack methods in Table I.

TABLE II
THE SUCCESS RATE OF ADVERSARIAL EXAMPLES

Method \ Type	Targeted	Non-targeted
L-BFGS [22]	96.1%	NA
FGSM [25]	NA	74.6%
IGSM [32]	85.4%	99.1%
MIM [47]	99.3%	100%
DeepFool [44]	NA	100%
CW Attack [46]	93.2%	98.4%
Uni.perturbations [45]	NA	88.2%
JSMA* [30]	-	-

* Out of Memory.

V. COMPARISON OF VARIOUS ATTACK METHODS

In this section, we compare the attributes of different attack methods in terms of black/white-box, attack type, targeted/non-targeted and PASS. Then we conduct a lot of experiments to compare the success rate and the transfer rate for different attack methods. The code to reproduce our experiments is available online at http://hardwaresecurity.cn.

A. Experimental Setup

Platform. All the experiments are conducted on a machine equipped with an AMD Threadripper 1920X CPU, a NVIDIA GTX 1050Ti GPU and 16G memory, and implemented in tensorflow-gpu 1.6.0 with Python 3.6.

Dataset. All of the experiments described in this section are performed on ILSVRC 2012 dataset [53]. The ILSVRC 2012 dataset is a subset of ImageNet, containing about 1000 images of 1000 categories. The examples are split among a training set of approximately 1.2 million examples, a validation set of 50,000 and a test set of 150,000. In our experiments, we choose 1000 images belonging to the 1000 categories randomly from the ILSVRC 2012 validation set. The size of

TABLE III
COMPARISON OF DIFFERENT ATTACK METHODS ON ATTRIBUTES

Method	Black/White-box	Attack Type	Targeted/Non-targeted	PASS
L-BFGS [22]	White-box	Gradient-based	Targeted	*
FGSM [25]	White-box	Gradient-based	Non-targeted	* * * * *
IGSM [32]	White-box	Gradient-based	Targeted/Non-targeted	**
Iter.l.l [32]	White-box	Gradient-based	Non-targeted	**
JSMA [30]	White-box	Gradient-based	Targeted	**
Uni.perturbations [45]	White-box	Decision boundary-based	Non-targeted	* * *
DeepFool [44]	White-box	Decision boundary-based	Non-targeted	*
One-pixel [50]	Black-box	-	Non-targeted	* * * *
CW Attack [46]	White-box	Iterative optimization	Targeted/Non-targeted	*
Ensemble Attack [38]	White-box	Ensemble optimization	Non-targeted	* * *

TABLE IV
THE TRANSFER RATE OF ADVERSARIAL EXAMPLES

Method \ Model	Targeted/Non-targeted	A→B	A→C	A→D	A→E	Perturbations(P)
L-BFGS [22]	Targeted	0.3%	0.3%	0.3%	0.2%	1.02
FGSM [25]	Non-targeted	66.8%	51.9%	51.5%	64.8%	15.23
IGSM [32]	Targeted	0.3%	0.2%	0.1%	0.1%	8.87
	Non-targeted	39.9%	34.4%	33.7%	46.7%	8.85
MIM [47]	Targeted	0%	0.4%	0.3%	0.5%	13.55
	Non-targeted	56.4%	45.4%	44.8%	59.9%	13.54
DeepFool [44]	Non-Targeted	30.5%	16.6%	10.5%	16.7%	0.49
CW Attack [46]	Targeted	0%	0%	0%	0%	0.78
	Non-targeted	4.3%	3.9%	2.9%	5.3%	0.76
Uni.perturbations [45]	Non-targeted	44.2%	29.1%	23.2%	35.4%	6.01
JSMA* [30]	Targeted	-	-	-	-	-

* Out of Memory.

each image is $299 \times 299 \times 3$. The intensity values of pixels in all these images are scaled to a real number in [0, 1].

DNN Model. For the dataset, we evaluate the success rate and transfer rate of AEs on five popular deep network architectures: Incpetion V3 [54], AlexNet [55], ResNet34 [56], DenseNet20 [57] and VGG19 [58], which are pretrained by the ImageNet dataset.

B. Comparison of Different Attack Methods on Attributes

Table III summarizes the attribute information of mainstream attacks. We find that, 1) most of the attack methods are white-box attacks. In the scenario of black-box attacks, attackers are difficult to construct AE attacks on machine learning models; 2) most of mainstream attacks are based on the gradient. Later on, other attacks such as decision boundary, iterative optimization and ensemble optimization are proposed; 3) the more asterisks mean the larger value of PASS, i.e., the generated AEs are easier to be perceived by the human eye. For example, since FGSM is prone to the label leakage effect,

the best perturbation to generate AEs is large (i.e., the PASS is large) and hence the generated AEs are easier to be perceived.

C. The Success Rate

The success rate of the various AE methods for targeted and non-targeted attacks on the Inception V3 model are evaluated. As shown in Table II, the success rate of targeted attacks is lower than the non-targeted attacks for the same AE method, and MIM has highest success rate for target attacks and non-target attacks. Note that JSMA needs to calculate the forward derivative of each pixel in the image to construct the Jacobian matrix. Therefore, JSMA is computationally expensive. In the experiment, we do not report the success rate and transfer rate for JSMA because it runs out of memory on the big dataset.

D. The Transfer Rate

The transfer rates of typical AEs for targeted and non-targeted attacks among the five models (Incpetion V3 (A),

AlexNet (B), ResNet (C), DenseNet (D) and VGG (E)) are shown in Table IV. Incpetion V3 is used as the source model and the other models are used as the target models. We collect 1000 adversarial images for each AE method generated in Incpetion V3 and apply them to other models to test the transfer rate of AEs. For each AE method, we define the average perturbation P as follows:

$$P = \frac{\sum_{i=1}^{N} \sum_{r=1}^{R} \sum_{c=1}^{C} \sum_{l=1}^{L} abs(pixel_{i,r,c,l}^{ori} - pixel_{i,r,c,l}^{adv})}{N \times R \times C \times L} \tag{24}$$

where N, R, C and L are the number of images, rows, columns and layers respectively, $pixel_{i,r,c,l}^{ori(adv)}$ is the pixel value of row r, column c, and layer l of original adversarial image i. $abs(pixel_{i,r,c,l}^{ori} - pixel_{i,r,c,l}^{adv})$ is the absolute value of $(pixel_{i,r,c,l}^{ori} - pixel_{i,r,c,l}^{adv})$. The magnitude of the perturbation is proportional to the transfer rate, i.e., the greater the perturbation, the higher the transfer rate. In addition, we find that the transfer rate of AE methods is low, i.e., the transfer rates of most of targeted attacks are less than 0.5%, which means that it is difficult to conduct AE attacks in the real black-box scenario and it is urgent to develop AE attack methods with the high transfer rate.

VI. DEFENSES AGAINST ADVERSARIAL EXAMPLES

AEs bring a great threat to the security-critical AI applications such as face payment [60], medical systems [61] and autonomous vehicles [62], [63] based on image recognition in deep learning. Vulnerability to AEs is not unique to deep learning. All machine learning models are vulnerable to AEs [28]. Therefore, defending against AEs is urgent for machine learning security. In this section, we will briefly describe the basic goals of defending against AEs, then detail the current defense techniques and their limitations. Finally, some suggestions are presented for future research work on the problems of the current defense techniques.

A. Defense Goals

Generally, there are four defense goals:

1) **Low impact on the model architecture:** when constructing any defense against AEs, the primary consideration is the minimal modification to model architectures.

2) **Maintain model speed:** running time is very important for the availability of DNNs. It should not be affected during testing. With the deployment of defenses, DNNs should still maintain high performance on large datasets.

3) **Maintain accuracy:** defenses should have little impact on the classification accuracy of models.

4) **Defenses should be targeted:** defenses should be effective on the examples which are relatively close to the training set. Since the examples that are far from the dataset are relatively secure, the perturbations to these examples are easily detected by the classifier.
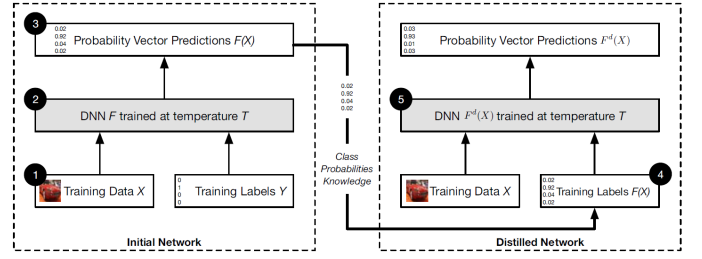

Fig. 12. **The pipeline of defensive distillation.** The initial network is trained at temperature T on the training set $(X,Y(X))$, the distilled network is trained at the same temperature T on the new training set $(X,F(X))$ [59].

B. Current Defenses

1) Adversarial Training

AEs have been used to improve anti-interference ability for AI models. In 2015, Goodfellow et al. [25] proposed the adversarial training to improve the robustness of the model. The basic idea is to add AEs into the training data to continuously generate new AEs at each step of the training. The number and relative weight of AEs in each batch are controlled by the loss function independently. The corresponding loss function is as follows:

$$LOSS = \frac{1}{(m-k)+\lambda k}\left(\sum_{i \in CLEAN} L(X_i|y_i) + \lambda \sum_{i \in ADV} L(X_i^{adv}|y_i)\right) \tag{25}$$

where $L(X|y)$ is a loss function of the example X with a true label y, m is the total number of training examples, k is the number of AEs, and λ is a hyper-parameter used to control the relative weight of the AEs in the loss function. When $k = 0.5m$, i.e., when the number of AEs is the same as the number of original examples, the model has the best effect in the adversarial training.

Adversarial training is not the same as data augmentation. The augmented data may appear in the test set, while AEs are usually not shown in the test set but can reveal the defects of the model. Adversarial training can be viewed as the process of minimizing classification error rates when the data is maliciously perturbed. In the following two situations, we suggested to use adversarial training:

i) **Overfitting:** when a model is overfitting, a regularization term is needed.

ii) **Security:** when AEs refer to security problems, adversarial training is the most secure method among all known defenses with only a small loss of accuracy.

Although a model is robust to white-box attacks after adversarial training, it is still vulnerable to the AEs generated from other models, i.e., the model is not robust to black-box attacks. Based on this attribute, Tramr et al. [49] proposed the concept of ensemble adversarial training. The main idea is to augment the training data which constructed not only from the model being trained but also from the other pre-trained models, which increases the diversity of AEs and improves the generalization ability.

2) Defensive Distillation

Adversarial training needs AEs to train the model, thus the defense is related to the process of AEs construction. For any defense, the defense effect is quite different for different attack methods. In 2016, Papernot et al. [59] proposed a universal defensive method for neural networks, which is called defensive distillation. The distillation method uses a small model to simulate a large and computationally intensive model, which does not affect the accuracy and can solve the problem of information missing. Different from the traditional distillation technique, defensive distillation aims to smooth the model during the training process by generalizing examples outside the training data. The specific training steps are shown in Fig. 12.

i) The probability vectors produced by the first DNN are used to label the dataset. These new labels are called soft labels as opposed to hard class labels.

ii) The newly labelled dataset is used to train the second DNN model. The second model can also be trained by using a combination of the hard and soft labels. Since the second model combines the knowledge of the first model, the size of the second model is smaller in scale, less computationally expensive and more robust than the first model.

The basic idea of defensive distillation is to generate smooth classifiers that are more resilient to AEs, reducing the sensitivity of the DNN to the input perturbation. In addition, defensive distillation improves the generalization ability because it does not modify the neural network architecture. Therefore, it has low training overhead and no testing overhead. Although attack methods [35], [46] have demonstrated that the defensive distillation does not improve the robustness of neural networks significantly, it is still a good research direction to defense AEs following three ways:

(i) Consider defensive distillation under different types of the perturbation (FGSM, L-BFGS, etc.);

(ii) Investigate the effect of distillation on other DNN models and AE constructing algorithms;

(iii) Study various distance metrics such as L_0, L_2, L_∞ between the original examples and the AEs.

3) *Detector*

Adversarial training is proposed to enhance the robustness of the model. However, this method lacks generalization ability and is difficult to popularize. Defensive distillation is proposed to defend AEs but defeated by a strong CW Attack. In 2017, Lu et al. [64] proposed an RBF-SVM based detector to detect whether the input is normal or adversarial (as shown in Fig. 13). The detector can get the internal state of some back layers in the original classification neural network. If the detector finds out that the example is adversarial example, then it will be rejected.

We assume that the detector is difficult to be attacked, and the output of ReLU activation function is processed in the binary format. Since normal examples and AEs generate different binary codes, detectors can compare the code during the test to determine whether the input is normal or adversarial.

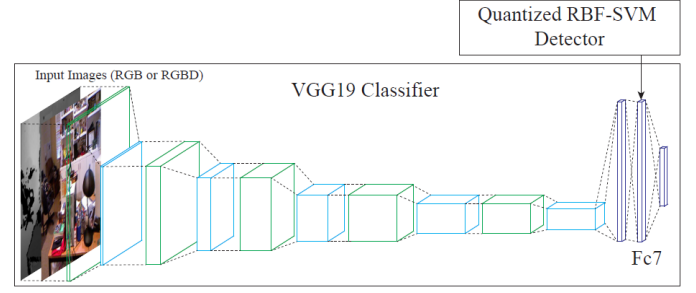At present, AE detectors are mainly divided into the following classes [65]:

Fig. 13. **SafeNet architecture.** SafetyNet consists of a conventional classifier with an RBF-SVM that uses discrete codes to detect adversarial examples [64].

Detection Based on Secondary Classification. Generally, there are two kinds of secondary classification detection methods. The first one is adversarial training detector [66], [67], which is similar to adversarial training. The main idea is to add a new classification label to AEs during training. If an AE is detected, the model will classify it into a new class. The second one is to take the characteristics extracted from AEs and original examples during the convolution layer as input. Then the labelled input data is used to train neural network detectors. This method performed well on detecting over 85% of AEs.

Detection Based on Principal Component Analysis (PCA). The essence of PCA is to transform the original features linearly and map them to a low-dimensional space with the best possible representation of the original features. PCA-based detection methods are mainly divided into two types. The first one uses PCA in the input layer due to the greater weight of AEs processed by PCA than original examples [68]. The second one uses PCA in the hidden layer [69]. If the result of each hidden layer matches the feature of original examples, the detector will classify the input as original examples.

Detection Based on Distribution. There are two main distribution-based detection methods. The first one uses the maximum mean discrepancy [66] which measures the distance between two different but related distributions. Assuming that there are two sets of images S_1 and S_2, S_1 contains all the original examples, S_2 contains either all AEs or all original examples. If S_1 and S_2 have the same distribution, then S_2 has original examples; otherwise, S_2 is full of AEs. The second one uses kernel density estimation [70]. Since AEs have a different density distribution from the original examples, they can be detected with high confidence by the estimation of the density ratio. If the density ratio of one example is close to 1, it belongs to original example. If the density ratio is much larger than 1, it belongs to AEs.

Other Detection Methods. Dropout randomization [29] is a method to use dropout randomly during AE detection. Original examples always generate correct labels, but AEs are of high possibility to be different from the label corresponding to original examples. In addition, another method called Mean Blur [69] uses the filter to perform mean blurring on the input image and can effectively improve the robustness of models.

TABLE V
THE ADVANTAGES AND DISADVANTAGES OF DEFENSES

Type	Method	Advantages	Disadvantages
Modified training/input	Adversarial training [25]	Simple, good defensive ability	Difficult to converge, high overhead
	Data compression [72]		
	Data randomization [73]		
Modifying the network	Deep Contractive Networks [74]	Low overhead, good generalization	Model-dependent, high complexity
	Gradient masking [75]		
	Defensive distillation [59]		
Network add-ons	GAN-based [76]	Low complexity, model independent	Weak generalization, not improving the robustness
	Detection [64]		

C. Other Defense Techniques

Defenders' goal is to train a model where no AEs exist or AEs cannot be easily generated. Recently, some novel researches on defending AEs have been proposed. Meng et al. [42] proposed a framework MagNet, including one or more separate detector networks and one reformer network. The detector network learns to distinguish normal examples from AEs by approximating normal examples. The reformer network moves AEs towards the normal examples. As MagNet is independent of the process of constructing AEs, hence it is effective in the black-box and gray-box attacks. Dong et al. [47] proposed a high-level representation guided denoiser method which requires fewer training images and consumes less training time than previous defense methods at the expense of low success rate reduction. Ma et al. [71] proposed Local Intrinsic Dimensionality (LID) to describe the dimensional attributes of the adversarial subspace in the AEs and proved that these features can distinguish normal examples from AEs effectively. Baluja et al. [77] proposed adversarial transformation networks (ATNs) to increase the diversity of perturbations to improve the effectiveness of adversarial training. However, ATNs may produce similar perturbations to iterative methods which are not suitable for adversarial training.

D. Limitations of Defenses

As discussed above, a lot of defenses have been proposed. In what follows, we summarize their advantages and disadvantages.

As shown in Table V, adversarial training is simple and can significantly improve the robustness of models. However, AEs are required in the training process, which brings high overhead. Besides, it is difficult to theoretically explain which attack method to construct AEs for adversarial training can achieve the best robustness of models. Defensive distillation can greatly reduce the sensitivity to perturbations without modifying the neural network architectures. Therefore, defensive distillation incurs low overhead in training and testing. However, defensive distillation needs to add distillation temperature and modify the objective function, which increases the complexity of designing defensive models. Besides, attackers can easily bypass the defensive distillation by the following three strategies: 1) choose a more suitable objective function; 2) calculate the final layer of gradient instead of the second-to-last layer of gradient; 3) attack a fragile model and then transfer to the distillation model. Detectors do not need to modify the model architecture and parameters, hence

the complexity is low. However, its performance is highly correlated with the type of detector. In addition, this method only detects the existence of AEs and does not improve the robustness of the model.

VII. RECENT CHALLENGES AND NEW OPPORTUNITIES

AE construction and defense are one of the research hotspots in the AI security field. Although many AE construction methods and defense techniques have been proposed, various unresolved problems still exist. This section summarizes the challenges to this field and put forward to some future research directions.

In term of AE construction, there are three major challenges:

1) **It is difficult to build a generalized AE-construction method.** In recent years, a lot of AE-construction methods have been proposed such as the gradient-based FGSM, JSMA, the decision boundary-based DeepFool and the ensemble attack method combining multiple models. These methods can achieve good performance in some evaluation metrics, but they are difficult to construct a generalized AE. Therefore, defenders can propose the efficient defenses against these specific attacks. For example, the gradient can be hidden or obfuscated to prevent against the gradient-based AE-construction methods.

2) **It is difficult to control the magnitude of perturbation for target images.** In the mainstream attack methods, attackers construct AEs by perturbing target images to fool neural network models. However, it is difficult to control the magnitude of perturbations because too small perturbations can not generate AEs and too large perturbations can be perceived by human eyes easily.

3) **AEs are difficult to maintain adversarial stability in real-world applications.** The image perturbed at specific distances and angles may lead to the model misclassification. However, a lot of images perturbed at different distances and angles fail to fool the classifier [78]. Moreover, AEs may lose its adversarial with physical transformation such as blurring, rotation, scaling and illumination [79]. Actually, it is hard for AEs to maintain stability in real-world applications.

Therefore, to address these issues, we propose to improve AE quality in the following three directions.

1) **Construct AEs with a high transfer rate.** With the diversification of neural network models, the effectiveness of attacks for single model is not enough. Based on the transferability, constructing AEs with high transfer rate is a prerequisite to evaluate the effectiveness of black-box attacks and a key metric to evaluate generalized attacks.

2) Construct AEs without perturbing the target image. When constructing AEs, the magnitude of perturbations to the target image is determined by experiments. Hence, the optimal perturbation will be different in various models. It increases the complexity of attacks and affects the success rate and the transfer rate. Therefore, constructing AEs without perturbing the target image is a novel and challenging research direction.

3) Model the physical transformation. In the physical world, attackers need to consider not only the magnitude of the perturbations but also the physical transformations such as translation, rotation, brightness, and contrast. However, it is difficult for attackers to use traditional algorithms to generate real-world AEs with high adversarial stability. Therefore, modeling physical perturbations is an efficient way to improve the stability of real-world AEs.

In terms of defending against AEs, there are two main challenges at present.

1) Defense is highly related to model architectures and parameters. The black-box attack does not need to obtain model architecture and parameters to construct AEs. Therefore, it is difficult for defenders to resist the black-box attack by modifying the model architectures or parameters. For example, defensive distillation needs to modify and retrain the target classifier.

2) Weak generalization for defense models. Adversarial training and detector are representative defense techniques. Adversarial training can improve the robustness of the model by adding AEs to train dataset. Detector can detect examples based on AEs in the dataset. However, the defense effect is quite different when defending AEs generated by different attack methods, i.e., the generalization ability of defense models is weak.

VIII. Conclusion

DNNs have recently achieved state-of-the-art performance on a variety of pattern recognition tasks. However, recent researches show that DNNs, like many other machine learning models, are vulnerable to AEs. Although many AE construction and defense methods have been proposed, there are still some challenges to be solved. The state-of-the-art research is still in the adversarial development stage of "while the priest climbs a post, the devil climbs ten". In this survey, we review the state-of-the-art AE construction methods and the corresponding defense techniques, then summarize several challenges along with the future trends in this field. Although AEs have caused the deep learning to be questioned, it also prompts both academia and industry to understand the difference between AI and our human brain better.

REFERENCES

[1] "The Brain (game show)," [Online]. Available: https://en.wikipedia.org/wiki/The Brain (game show), 2017.

[2] "AlphaGo Zero: Learning from scratch," [Online]. Available: https://deepmind.com/blog/alphago-zero-learning-scratch/, 2017.

[3] A. Bundy, "Preparing for the future of Artificial Intelligence," *AI & SOCIETY*, vol. 32, no. 2, pp. 285-287, 2017.

[4] L. E. Parker, "Creation of the National Artificial Intelligence Research and Development Strategic Plan," *AI Magazine*, vol. 39, no. 2, pp. 25-48, 2018.

[5] "A Next Generation Artificial Intelligence Development Plan: China," [Online]. Available: https://chinacopyrightandmedia.wordpress.com/2017/07/20/a-next-generation-artificial-intelligence-development-plan/.

[6] "Chubby robot goes haywire, injures human at trade show," [Online]. Available: https://www.dailydot.com/debug/first-robot-human-injury-china/, 2016.

[7] "Danger, danger! 10 alarming examples of AI gone wild," [Online]. Available: https://www.infoworld.com/article/3184205/technology-business/danger-danger-10-alarming-examples-of-ai-gone-wild.html, 2017.

[8] "Uber self-driving car kills Arizona pedestrian, realizing worst fears of the new tech," [Online]. Available: https://www.usatoday.com/story/tech/2018/03/19/uber-self-driving-car-kills-arizona-woman/438473002/.

[9] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognit.*, vol. 84, pp. 317-331, Dec. 2018.

[10] M. Brundage et al., "The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation," 2018, arXiv:1802.07228.

[11] G. L. Wittel and S. F. Wu, "On Attacking Statistical Spam Filters," in *Proceedings of the Conference on Email & Anti-spam*, 2004.

[12] B. Biggio, G. Fumera, F. Roli, and L. Didaci. "Poisoning adaptive biometric systems,"*Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*,vol. 7626 LNCS, 2012, pp. 417-425.

[13] B. Biggio, L. Didaci, G. Fumera, and F. Roli, "Poisoning attacks to compromise face templates," in *Proceedings of the International Conference on Biometrics*, 2013, pp. 1-7.

[14] B. Biggio et al., "Evasion Attacks against Machine Learning at Test Time," in *Proceedings of the 2013th European Conference on Machine Learning and Knowledge Discovery in Databases*, 2013, pp. 387-402.

[15] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial Examples for Malware Detection," in *European Symposium on Research in Computer Security*, 2017, pp. 62-79.

[16] Z. Abaid, M. A. Kaafar, and S. Jha, "Quantifying the impact of adversarial evasion attacks on machine learning based android malware classifiers," in *IEEE International Symposium on Network Computing and Applications (NCA)*, 2017, pp. 1-10.

[17] D. Lowd and C. Meek, "Adversarial learning," *in Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in Data Mining*, 2005, pp. 641-647.

[18] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D.Tygar, "Can machine learning be secure? (Invited Talk)," *in Proceedings of ACM Symposium on Information, Computer and Communications Security*, 2006, pp. 16-25.

[19] B. I. P. Rubinstein et al., "ANTIDOTE: understanding and defending against poisoning of anomaly detectors," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, 2009, pp.1-14.

[20] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma, "Adversarial classification," *in Proceedings of the 2004 ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, 2004, pp. 99-108.

[21] F. Zhang, P. P. K. Chan, B. Biggio, D. S. Yeung, and F. Roli, "Adversarial Feature Selection Against Evasion Attacks," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 766-777, Mar. 2016.

[22] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna,D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *in Proceedings of International Conference on Learning Representations (ICLR)*, 2014, pp. 1-10.

[23] "Even Artificial Neural Networks Can Have Exploitable 'Backdoors'", [Online]. Available at: https://www.wired.com/story/machine-learning-backdoors/

[24] H. Chen, H. Zhang, P.-Y. Chen, J. Yi, and C.-J. Hsieh, "Attacking Visual Language Grounding with Adversarial Examples: A Case Study on Neural Image Captioning," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018, pp. 2587-2597.

[25] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," 2014, arXiv:1412.6572.

[26] Q. Zheng, M. Yang, J. Yang, Q. Zhang, and X. Zhang, "Improvement of Generalization Ability of Deep CNN via Implicit Regularization in Two-Stage Training Process," *IEEE Access*, vol. 6, pp. 15844-15869, 2018.

[27] G. Philipp and J. G. Carbonell, "The Nonlinearity Coefficient - Predicting Overfitting in Deep Neural Networks," 2018, arXiv:1806.00179

[28] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha,Z. B. Celik, and A. Swami, "Practical Black-Box Attacks against Machine Learning," *in Proceedings of ACM Asia Conference on Computer and Communications Security*, 2017, pp. 506-519.

[29] I. Sutskever, G. Hinton, A. Krizhevsky, and R. R.Salakhutdinov, "Dropout : A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929-1958, 2014.

[30] N. Papernot, P. Mcdaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," *in Proceedings of IEEE European Symposium on Security and Privacy (EuroS&P)*, 2016, pp. 372-387.

[31] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial Machine Learning at Scale," *in Proceedings of IEEE European Symposium on Security and Privacy (EuroS&P)*, 2016, pp. 1-17.

[32] A. Kurakin and I. Goodfellow, "Adversarial examples in the physical world," *in Proceedings of International Conference on Learning Representations (ICLR)*, 2016, pp. 1-14.

[33] D. Su, H. Zhang, H. Chen, J. Yi, P. Y. Chen, and Y. Gao, "Is robustness the cost of accuracy? A comprehensive study on the robustness of 18 deep image classification models," *in Proceedings of European Conference on Computer Vision (ECCV)*, 2018, pp. 644-661.

[34] A. Rozsa, M. Gunther, and T. E. Boult, "Are accuracy and robustness correlated?" *in Proceedings of the 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2016, pp. 227-232.

[35] N. Carlini and D. Wagner, "Defensive Distillation is Not Robust to Adversarial Examples," 2016, arXiv:1607.04311.

[36] A. Fawzi, O. Fawzi, and P. Frossard, "Analysis of classifiers robustness to adversarial perturbations," *Machine Learning*, vol. 107, no. 3, pp. 481-508, 2018.

[37] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "The Space of Transferable Adversarial Examples," 2017, arXiv:1704.03453.

[38] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into Transferable Adversarial Examples and Black-box Attacks," 2016, arXiv:1611.02770.

[39] L. Wu, Z. Zhu, C. Tai, and W. E, "Understanding and Enhancing the Transferability of Adversarial Examples," 2018, arXiv:1802.09707.

[40] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612, 2004.

[41] A. Rozsa, E. M. Rudd, and T. E. Boult, "Adversarial Diversity and Hard Positive Generation," *in Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2016, pp. 410-417.

[42] D. Meng and H. Chen, "MagNet: a Two-Pronged Defense against Adversarial Examples," *in Proceedings of ACM SIGSAC Conference on Computer and Communications Security, New York*, 2017, pp. 135-147.

[43] N. Narodytska and S. Kasiviswanathan, "Simple Black-Box Adversarial Attacks on Deep Neural Networks," *in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 1310-1318.

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

[45] S. M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," *in Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 86-94.

[46] N. Carlini and D. Wagner, "Towards Evaluating the Robustness of Neural Networks," *in Proceedings of IEEE Symposium on Security and Privacy (EuroS&P)*, 2017, pp. 39-57.

[47] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting Adversarial Attacks with Momentum," *in Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 9185-9193.

[48] F. Xia and R. Liu, "Adversarial Examples Generation and Defense Based on Generative Adversarial Network," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 3905-3911, 2018.

[49] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow,D. Boneh, and P. McDaniel, "Ensemble Adversarial Training: Attacks and Defenses," in *Proceedings of the International Conference on Learning Representations*, 2018.

[50] J. Su, D. V. Vargas, and K. Sakurai, "One Pixel Attack for Fooling Deep Neural Networks," 2017, arXiv:1710.08864.

[51] 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," in *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018, pp. 1-18.

[52] Brown et al., "Adversarial Patch", in *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS)*, 2017, Long Beach, CA, USA.

[53] O. Russakovsky et al., "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211-252, 2015.

[54] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818-2826.

[55] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Convolutional Neural Networks ImageNet Classification with Deep Convolutional Neural Network," *Communications of the ACM*, vol. 60, no. 06, 2017.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770-778.

[57] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261-2269.

[58] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *in Proceedings of International Conference on Learning Representations*, 2015.

[59] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks," *in Proceedings of IEEE Symposium on Security and Privacy*, 2016, pp. 582-597.

[60] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Adversarial Generative Nets: Neural Network Attacks on State-of-the-Art Face Recognition," 2017, arXiv:1801.00349.

[61] S. G. Finlayson, I. S. Kohane, and A. L. Beam, "Adversarial Attacks Against Medical Deep Learning Systems." arXiv:1804.05296, 2019.

[62] Y. Huang and S. H. Wang, "Adversarial Manipulation of Reinforcement Learning Policies in Autonomous Agents," *in Proceedings of International Joint Conference on Neural Networks*, 2018, pp. 1-8.

[63] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, and P. Abbeel, "Adversarial Attacks on Neural Network Policies," 2017, arXiv:1702.02284.

[64] J. Lu, T. Issaranon, and D. Forsyth, "SafetyNet: Detecting and Rejecting Adversarial Examples Robustly," *in Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 446-454.

[65] N. Carlini and D. Wagner, "Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods," 2017, arXiv:1705.07263.

[66] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. McDaniel, "On the (Statistical) Detection of Adversarial Examples," 2017, arXiv:1702.06280.

[67] Z. Gong, W. Wang, and W.s. Ku, "Adversarial and Clean Data Are Not Twins," 2017, arXiv:1704.04960.

[68] D. Hendrycks and K. Gimpel, "Early Methods for Detecting Adversarial Images," *in Proceedings of International Conference on Learning Representations (ICLR)*, 2016, pp. 1-9.

[69] X. Li and F. Li, "Adversarial Examples Detection in Deep Networks with Convolutional Filter Statistics," *in Proceedings of IEEE International Conference on Computer Vision*, 2017, pp. 5775-5783.

[70] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting Adversarial Samples from Artifacts," 2017, arXiv:1703.00410.

[71] X. Ma, B. Li, Y. Wang, S. M. Erfani, S. Wijewickrema,G. Schoenebeck, D. Song, M. E. Houle, and J. Bailey, "Characterizing Adversarial Subspaces Using Local Intrinsic Dimensionality," 2018, arXiv:1801.02613.

[72] G. K. Dziugaite and D. M. Roy, "A study of the effect of JPG compression on adversarial images" 2016, arXiv: 1608.00853

[73] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie and A. Yuille, "Adversarial Examples for Semantic Segmentation and Object Detection," *in Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1378-1387.

[74] S. Gu and L. Rigazio, "Towards Deep Neural Network Architectures Robust to Adversarial Examples," *in Proceedings of International Journal on Computational Science & Applications*, vol. 4, no. 2013, pp. 1-9, 2014.

[75] A. S. Ross and F. Doshi-velez, "Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing their Input Gradients," in *Proceedings of AAAI Conference on Artificial Intelligence*, 2018.

[76] H. Lee, S. Han, and J. Lee, "Generative Adversarial Trainer: Defense to Adversarial Perturbations with GAN," 2017, arXiv:1705.03387

[77] Baluja, Shumeet , and I. Fischer . "Adversarial Transformation Networks: Learning to Generate Adversarial Examples," 2017, arXiv:1703.09387.

[78] J. Lu, H. Sibai, E. Fabry, and D. Forsyth, "NO Need to Worry about Adversarial Examples in Object Detection in Autonomous Vehicles," 2017, arXiv:1707.03501 .

[79] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing Robust Adversarial Examples," 2017, arXiv:1707.07397.

[80] N. Akhtar and A. Mian, "Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey," *IEEE Access*, vol. 6, pp. 14410-14430, 2018.