Adversarial Machine Learning in Image Classification: A Survey Toward the Defender’s Perspective

GABRIEL RESENDE MACHADO, Military Institute of Engineering (IME), Brazil
EUGÊNIO SILVA, State University of West Zone (UEZO), Brazil
RONALDO RIBEIRO GOLDSCHMIDT, Military Institute of Engineering (IME), Brazil

Deep Learning algorithms have achieved state-of-the-art performance for Image Classification. For this reason, they have been used even in security-critical applications, such as biometric recognition systems and self-driving cars. However, recent works have shown those algorithms, which can even surpass human capabilities, are vulnerable to adversarial examples. In Computer Vision, adversarial examples are images containing subtle perturbations generated by malicious optimization algorithms to fool classifiers. As an attempt to mitigate these vulnerabilities, numerous countermeasures have been proposed recently in the literature. However, devising an efficient defense mechanism has proven to be a difficult task, since many approaches demonstrated to be ineffective against adaptive attackers. Thus, this article aims to provide all readerships with a review of the latest research progress on Adversarial Machine Learning in Image Classification, nevertheless, with a defender’s perspective. This article introduces novel taxonomies for categorizing adversarial attacks and defenses, as well as discuss possible reasons regarding the existence of adversarial examples. In addition, relevant guidance is also provided to assist researchers when devising and evaluating defenses. Finally, based on the reviewed literature, this article suggests some promising paths for future research.

CCS Concepts:
• Information systems $\rightarrow$ Decision support systems;
• Security and privacy $\rightarrow$ Domain-specific security and privacy architectures;
• Computing methodologies $\rightarrow$ Neural networks;

Additional Key Words and Phrases: Computer vision, image classification, adversarial images, deep neural networks, adversarial attacks, defense methods

ACM Reference format:
Gabriel Resende Machado, Eugênio Silva, and Ronaldo Ribeiro Goldschmidt. 2021. Adversarial Machine Learning in Image Classification: A Survey Toward the Defender’s Perspective. *ACM Comput. Surv.* 55, 1, Article 8 (November 2021), 38 pages.
https://doi.org/10.1145/3485133

1 INTRODUCTION

In recent years, Deep Learning algorithms have made important and rapid progress in solving numerous tasks involving complex analysis of raw data. Among some relevant cases are major
advances in speech recognition and natural language processing [36, 189], games [154], financial market analysis [73], fraud and malware detection [35, 89], prevention of DDoS attacks [196], and Computer Vision [69, 77, 90, 164, 165]. In the field of Computer Vision, Convolutional Neural Networks (CNNs) became the state-of-the-art Deep Learning algorithms since Krizhevsky et al. [90] presented innovative results in image classification tasks using the AlexNet architecture. Thereafter, motivated by the continuous popularization of GPUs and frameworks, CNNs kept growing in performance, currently even used in security-critical applications, such as medical sciences and diagnostics [104], autonomous vehicles [13], surveillance systems [40], and biometric and handwritten characters recognition [161, 170].

Nevertheless, some researchers began to argue if the same deep learning algorithms, which could even surpass the human performance [87], were actually robust enough to be used in safety-critical environments. Unfortunately, since the paper of Szegedy et al. [166], various other works have also highlighted the vulnerability of deep learning models in different tasks, such as speech recognition [23], text classification [45], malware detection [65], and specially image classification [22, 62, 138], before adversarial attacks. Adversarial attacks are usually conducted in the form of subtle perturbations generated by an optimization algorithm and inserted into a legitimate image to produce an adversarial example that, in the field of Computer Vision, is specifically known as adversarial image. After being sent to be classified, an adversarial image is often able to lead CNNs to produce a prediction different from the expected, usually with high confidence. Adversarial attacks on image classifiers are the most common in the literature and, for this reason, are the focus of this article.

The vulnerability of CNNs and other Deep Learning algorithms to adversarial attacks pushed the scientific community to revisit all the processes related to the construction of intelligent models, as an attempt to hypothesize some possible reasons concerning this lack of robustness. This arms race between attacks and defenses against adversarial examples ended up forming a recent research area called Adversarial Machine Learning that, in a nutshell, struggles to construct more robust Deep Learning models.

Adversarial Machine Learning in Image Classification is currently a very active research path that is responsible for most of the work in the area, with novel papers produced almost daily. However, there is neither a known efficient solution for securing Deep Learning models nor any fully accepted explanations for the existence of adversarial images yet. Taking into account the dynamism and relevance of this research area, it is crucial to be available in literature comprehensive and up-to-date review papers to position and orientate their readers about the current scenario. Although there are already some extensive surveys [2, 183, 195], they have already become somewhat outdated due to the great activity in the area. Moreover, they bring out a general overview of the Adversarial Machine Learning field that are not focused enough on works that propose defenses against adversarial attacks.

Therefore, keeping in mind the importance of Adversarial Machine Learning in Image Classification toward the development of more robust defenses and architectures against adversarial attacks, this article aims to provide all readerships an exhaustive review of the literature, however with a defender’s perspective. This survey covers the background needed to clarify to the reader the essential concepts in the area, as well the technical formalisms related to adversarial examples and attacks. Furthermore, a comprehensive survey of defenses and countermeasures against adversarial attacks is made and categorized on a novel taxonomy. Then, based on the works of Carlini and Wagner [20] and Carlini et al. [18], the present article discusses some principles for designing

\*Robustness can be defined as the capacity of a model or defense to tolerate adversarial disturbances by delivering reliable and stable outputs [186].
and evaluating defenses, which are intended to guide researchers to introduce stronger security methods. Essentially, the main contributions of this work are the following:

- The update of some existing taxonomies to categorize different types of adversarial images and novel attack approaches that are raised in the literature;
- The discussion and organization of defenses against adversarial attacks based on a novel taxonomy;
- An overview of understudied and overstudied scenarios using the introduced taxonomies, as well as the discussion of promising research paths for future works.

The remainder of this article is structured as follows: Section 2 brings an essential background that covers important concepts for the proper understanding of this work. Section 3 formalizes and also categorizes adversarial examples and attacks. Section 4 makes a review of existing defenses in the literature and proposes a novel taxonomy for organizing them. Section 5 addresses and formalizes relevant explanations for the existence of adversarial examples that supported the development of attacks and defenses. Section 6 provides close guidance based on relevant works to help defenders and reviewers to design and evaluate security methods, respectively. Section 7 lists promising research paths in Adversarial Machine Learning for Image Classification, and Section 8 brings the final considerations.

2 BACKGROUND

This article focuses mostly on CNNs, which are a specific type of Deep Neural Network and currently are the state-of-the-art algorithms for Image Classification [177]; however, other important learning models in Adversarial Machine Learning are next discussed. It is also worth mentioning that Table 1 lists the most important notations, along with their respective description, to provide the readers a convenient reference for the different symbols used throughout this survey.

2.1 Convolutional Neural Networks

Traditional CNN architectures usually perform feature learning by making use of (i) convolution and (ii) pooling layers on which, respectively, extracts useful features from images and reduces their spatial dimensionality. After feature learning, comes the fully connected layer (FC), which works similarly to an ordinary neural network. In a classification task, FCs produce a
single probability vector as output, called the \textit{probability vector}. The probability vector contains membership probabilities of a given input $x$ corresponding to each label $y_i \in Y_n$, where $Y_n$ is the set containing all the $n$ ground-truth labels of the original classification problem. The predicted label for $x$ is the one with the highest membership probability.

Some CNN winners of the ImageNet Challenge \cite{145}, such as AlexNet \cite{90}, ZF-Net \cite{198}, VGGNet \cite{156}, GoogLeNet \cite{164}, ResNet \cite{70}, and SE-Net \cite{77} stood out in literature over the years as the state-of-the-art architectures on image classification and recognition tasks. The increasing performance though came with the cost of more expensive computations and higher consumption of hardware resources, which are the main constraints that prevent the adoption of these models on mobile systems and IoT devices \cite{155}. Therefore, recent works are now investigating alternatives to reduce architectures’ parameters and computational effort without compromising their performance, where it can be mentioned, for example, the advances made toward \textbf{Binarized/Quantized Neural Networks (BNNs)} \cite{10} and \textbf{Neural Ordinary Differential Equations (Neural ODEs)} \cite{110}.

\section{Autoencoders}

An autoencoder is a neural network that aims to approximate its output to an input sample, or in other words, it tries to approximate an input $x$ to its \textit{identity function} by generating an output $\hat{x}$ as similar as possible to $x$ from a compressed representation learned. Despite seeming a trivial task, the autoencoder is actually trying to learn the inner representations of the input, regarding the structure of data. Autoencoders are useful for two main purposes: (i) dimensionality reduction, retaining only the most important data features \cite{76,107,148}, and (ii) data generation process \cite{38}.

\section{Generative Adversarial Networks}

\textbf{Generative Adversarial Networks (GANs)} are a framework introduced by Goodfellow et al. \cite{61} for building \textit{generative models} $P_G$, which resembles the data distribution $P_{data}$ used in the training set. GANs can be used to improve the representation of data, to conduct \textit{unsupervised learning} and to even construct defenses against adversarial images \cite{61,195}. Some works also used GANs for other purposes, such as image-to-image translation and visual style transfer \cite{83,200}. The GANs are composed of two models (usually deep networks) trained simultaneously: a \textit{generator} $G$ and a \textit{discriminator} $D$. The generator receives an input $x$ and tries to generate an output $q$ from a probability distribution $P_G$. In contrast, the discriminator classifies $q$, producing a label that determines if $q$ belongs to the distribution $P_{data}$ (benign or real input) or $P_G$ (fake or adversarial input). In other words, the \textit{generator} $G$ is actually being trained to fool the classifier $D$. In this competing scenario, GANs are usually capable of generating data samples that look close to benign examples.

\section{Adversarial Images and Attacks}

Formally, an adversarial image can be defined as follows: let $f$ be a classification model trained with legitimate images (i.e., images that do not have any malicious perturbations), and let $x$ be a legitimate input image (where $x \in \mathbb{R}^{w \times h \times k}$, such that $w$ and $h$ are the dimensions of the image and $k$ is its amount of color channels). Then it is crafted, from $x$, an image $x'$, such that $x' = x + \delta x$, where $\delta x$ is the perturbation needed to make $x$ cross the decision boundary, resulting $f(x) \neq f(x')$ (see Figure 1(a)). The perturbation $\delta x$ can also be interpreted as a vector $\delta x$, where its magnitude $||\delta x||$ represents the amount of perturbation needed to translate the point represented by the image $x$ in the space beyond the decision boundary. Figure 1(b) illustrates a didactic example of inserting a perturbation $\delta x$ into a legitimate image $x$ on a 2D space. According to Cao and Gong \cite{17}, an
adversarial image is considered optimal if it satisfies two requirements: (i) if the perturbations inserted on this image are imperceptible to human eyes and (ii) if these perturbations are able to induce the classification model to produce an incorrect output, preferably with a high confidence.

3.1 Taxonomy of Adversarial Images

This section is based on the works of Barreno et al., Huang et al., Kumar and Mehta, Xiao, Yuan et al., and Brendel et al. [9, 14, 78, 91, 186, 195], to propose a broader taxonomy to adversarial images formed by three different axes, as depicted by Figure 2: (i) perturbation scope, (ii) perturbation visibility, and (iii) perturbation measurement. It is also worth pointing out that all the axes introduced by the taxonomy in Figure 2 are in bold font for comparative purposes.

3.1.1 Perturbation Scope. Adversarial images may contain individual-scoped perturbations or universal-scoped perturbations.

- Individual-scoped perturbations are the most common in literature. They are generated individually for each input image;
- Universal-scoped perturbations are image-agnostic perturbations, i.e., they are perturbations generated independently from any input sample. Nevertheless, the resulting adversarial example is often able to lead models to misclassification [121, 123]. Universal perturbations permit adversarial attacks to be conducted more easily in real-world scenarios, given the fact they are crafted just once to be inserted into any sample belonging to a certain dataset.
3.1.2 Perturbation Visibility. The efficiency and visibility of perturbations can be organized as follows:

- **Optimal perturbations**: these perturbations are imperceptible to human eyes, but are useful to lead deep learning models to misclassification, usually with high confidence on the prediction;
- **Indistinguishable perturbations**: these perturbations are also imperceptible to human eyes. However, they are insufficient to fool deep learning models;
- **Visible perturbations**: these perturbations, when inserted into an image, can fool deep learning models. However, they can also be easily spotted by humans [15, 86];
- **Physical perturbations**: these perturbations are designed outside the digital scope and physically added to real-world objects themselves [47]. Although that some works adapted physical perturbations to Image Classification [92], they are performed usually in tasks involving Object Detection [31, 47, 169] (see Appendix C);
- **Fooling images**: these perturbations corrupt images to the point of making them unrecognizable by humans. Nevertheless, the classification models believe these corrupted images actually belong to one of the classes from the original classification problem, sometimes assigning to them high confidence on the prediction [129]. Fooling images are also known as **rubbish class examples** [62];
- **Noise**: in contrast to the malicious nature of perturbations, noises are non-malicious or non-optimal corruptions that may be present or inserted into an input image. Here it can be cited the gaussian noise.

3.1.3 Perturbation Measurement. Given the fact that it is difficult to define a metric that measures the capability of human vision, the p-norms are most used to control the size and the amount of the perturbations that are inserted into a image [119]. The p-norm $L_p$ computes the distance $\|x - x'\|_p$ in the input space between a legitimate image $x$ and the resulting adversarial example $x'$, where $p \in \{0, 1, 2, \infty\}$. In Equation (1) is defined the p-norm when $p = 1$ (Manhattan Distance) and $p = 2$ (Euclidean Distance):

$$L_p = \sum |x - x'|^p.$$

When $p = 0$, we count up the number of pixels that were modified in a legitimate sample to generate the adversarial image. However, the $L_\infty$ measures the maximum difference among all pixels in the corresponding positions between two images. For $L_\infty$ norm, each pixel is allowed to be modified within a maximum limit of perturbation, without having any restriction for the number of modified pixels. Formally, $L_\infty = \max(|x_1 - x'_1|, |x_2 - x'_2|, \ldots, |x_n - x'_n|)$. Despite the norms of $p \in \{0, 1, 2, \infty\}$ being the most used when computing perturbations, there are some works that defined custom metrics, as can be seen in Table 2.

### 3.2 Taxonomy of Attacks and Attackers

This section is also based on the concepts and definitions extracted from the works of Akhtar and Mian, Barreno et al., Brendel et al., Kumar and Mehta, Xiao, and Yuan et al. [2, 9, 14, 91, 186, 195] to gather existing definitions into a broader taxonomy that organizes adversarial attacks and attackers. In the context of security, adversarial attacks and attackers can be defined under threat models. A threat model defines the conditions under which a defense is designed to provide security guarantees against certain types of attacks and attackers [18]. Basically, a threat model delimits (i) the knowledge an attacker has about the targeted classifier (such as its parameters and architecture), (ii) his goal with the adversarial attack, and (iii) how he will perform the adversarial attack. A threat model can be then classified into six different axes, as depicted by Figure 3:
Fig. 3. Taxonomy of adversarial attacks and attackers.

(i) attacker’s influence, (ii) attacker’s knowledge, (iii) security violation, (iv) attack specificity, (v) attack computation, and (vi) attack approach. As in Figure 2, all the axes introduced by the taxonomy in Figure 3 are in bold font for comparative purposes.

3.2.1 Attacker’s Influence. This axis defines how the attacker will control the learning process of deep learning models. According to Xiao [186], the attacker can perform two types of attack, taking into account his influence on the classification model: (i) causative or poisoning attacks and (ii) evasive or exploratory attacks.

- Causative or poisoning attacks: in causative attacks, the attacker influences the learning process of the deep learning model during its training stage. In this type of attack, the training samples can either get corrupted by the attacker or the training set can get polluted with adversarial examples to produce a classification model incompatible with the original data distribution;
- Evasive or exploratory attacks: in contrast to causative attacks, in evasive attacks the attacker influences the deep learning models during the inference or testing stage. Evasive attacks are the most common type of attack, where the attacker crafts adversarial examples that lead deep learning models to misclassification, usually with high confidence in the prediction. Evasive attacks can also have an exploratory nature, where the attacker’s objective is to gather information about the targeted model, such as its parameters, architectures, cost functions, and so on. The most common exploratory attack is the input/output attack, where the attacker provides the targeted model with adversarial images to observe the outputs given by this model and tries to reproduce a similar substitute or surrogate model. The input/output attack is usually the first step to perform black-box attacks (see Section 3.2.2).

3.2.2 Attacker’s Knowledge. Taking into consideration the attacker’s knowledge with respect to the targeted model, three types of attacks can be performed: (i) white-box attacks, (ii) black-box attacks, and (iii) grey-box attacks.

- White-box attacks: in a white-box attack, the attacker has full access to the model’s and even the defense’s parameters and architectures, whenever such defense exists. This attack scenario probably would be the least frequent in real-world applications, due to the adoption of protection measures (such as user control, for example) to prevent unauthorized people access to the system components. By contrast, white-box attacks are usually the most powerful type of adversarial attack, and for this reason, are commonly used to evaluate the robustness of defenses and/or classification models when they are undergone to harsh conditions. Unfortunately, elaborating countermeasures resistant to white-box attacks is, so far, an open problem;
Black-box attacks: in this scenario, the attacker neither has access nor knowledge about any information concerning the classification model and the defense method, when present. Black-box attacks impose more restrictions to attackers, nonetheless, they are important when reproducing external adversarial attacks aiming deployed models, which, in turn, better represent real-world scenarios [137]. Despite the greater difficulty to perform black-box attacks, the attacker still might be able to evade the target model due to the transferability of adversarial examples. Works such as Szegedy et al. and Papernot et al. [137, 166] demonstrated the malicious effect of an adversarial image, generated using a certain classifier, can transfer and fool other classifiers, including the ones created by different learning algorithms (check Section 5.7 for more details). With this property in favor of the attacker, it can create an empirical model through a causative attack called substitute or surrogate model \( \hat{f} \), which has similar parameters to the targeted model \( f \). Therefore, the attacker can use \( \hat{f} \) to craft adversarial images and, afterwards, deploy them to be, oftentimes, misclassified by the targeted model;

Grey-box attacks: this attack scenario was first proposed by Meng and Chen [119]. In grey-box attacks, the attacker has access to the classification model but does not have access to any information concerning the defense method. Grey-box attacks are an intermediate alternative to evaluate defenses and classifiers, since they impose a greater threat level when compared to the black-box attacks, but without giving a wide advantage to the attacker when providing it with all the information concerning the defense method, as performed in white-box scenarios.

3.2.3 Security Violation. Security violations are often associated with the attacker’s objective when performing an adversarial attack against a classifier. The security violations caused by adversarial attacks can affect the (i) integrity, (ii) availability, and the (iii) privacy of the targeted classifiers.

Integrity violation: occurs whenever an adversarial sample degrades the performance of a model without compromising its normal operation, i.e., the model does not have its performance on legitimate samples actually affected. Evasive attacks are an example of integrity violation. [78, 186];

Availability violation: happens when the performance of the model on legitimate samples is also significantly affected to the point of making this model unusable, and therefore causing a denial of service. This kind of violation usually comes through a poisoning attack [186];

Privacy violation: here, the attacker tries to gain knowledge about key information not intended to be shared, i.e., information regarding the model’s architecture, parameters, and even its training data. Some examples of privacy violation against learning models that can be done in adversarial settings are:

i. Model Inversion: aims to reconstruct the data used to train a model \( f \) by using its parameters or even its outputs, as is usually performed in black-box settings [194];

ii. Membership Inference: infers information about a model’s training dataset \( T \) by determining whether a certain input is indeed part of \( T \) [153];

iii. Reverse-engineering: occurs when the attacker aims to build a surrogate model \( \hat{f} \), which closely approximates the targeted model \( f \) [174]. Reverse-engineering attacks are usually adopted as a previous step when performing adversarial black-box attacks [137].

3.2.4 Attack Specificity. Regarding specificity, an attacker can perform a (i) targeted attack and an (ii) untargeted attack (also known as non-targeted attack or indiscriminate attack [2, 106]). Targeted attacks aim to craft adversarial images that can lead the model to misclassify in a
predetermined class, chosen beforehand by the attacker. However, untargeted attacks seeks to fool the model by just choosing any class different from the ground-truth class of the original example. Formally, let \( x \) be a legitimate image, \( y \) its ground-truth class, \( Y \) the label set, and \( f \) a classification model. Then, an adversarial image \( x' = x + \delta x \) is crafted from \( x \). In a targeted attack, the attacker seeks to produce an output \( y' = f(x') \) such that \( y' \in Y \), and \( y' \neq y \), where \( y' \) corresponds to the attacker’s aimed class. Conversely, in an untargeted attack, the attacker seeks to produce an output \( y' = f(x') \), such that \( y' \in Y' \), \( Y' \subset Y \), and \( Y - Y' = \{y\} \).

3.2.5 Attack Computation. The algorithms used to compute perturbations can be (i) one-step and (ii) iterative. One-step algorithms, also known as one-shot algorithms [2, 93], compute \( \delta x \) in a single step by using the gradients of the model’s loss with respect to the legitimate image \( x \) to find the direction in space (i.e., the most prominent pixels in the legitimate image), which will maximize the error when perturbed [182]. Iterative algorithms, in turn, make use of more iterations to craft and fine-tuning the perturbations, and for this reason, they have a higher computational cost when compared to one-step algorithms. However, the perturbations generated by iterative algorithms are usually smaller and more efficient to fool classification models than those generated by one-step procedures [2].

3.2.6 Attack Approach. Adversarial attacks can also be organized with respect to the approach used by the attack algorithm to craft the perturbation. The approach used by an adversarial attack can be based on (i) gradient, (ii) transferibility/score, (iii) decision, and (iv) approximation.

- Gradient-based attacks: this attack approach is the most used in literature. The gradient-based algorithms make use of detailed information of the target model concerning its gradient with respect to the given input. This attack approach is usually performed in white-box scenarios when the attacker has full knowledge and access to the targeted model;

- Transfer/Score-based attacks: these attack algorithms either depend on getting access to the dataset used by the targeted model or the scores predicted by it to approximate a gradient. Usually, the outputs obtained by querying a targeted deep neural network are used as scores. These scores are then used along with the training dataset to fit a surrogate model, which will craft the perturbations that will be inserted into the legitimate images. This attack approach is often useful in black-box attacks;

- Decision-based attacks: this approach was first introduced by Brendel et al. [14], and it is considered by the authors as a simpler and more flexible approach, since it requires few changes in parameters than gradient-based attacks. A decision-based attack usually queries the softmax layer of the targeted model and, iteratively, computes smaller perturbations by using a process of rejection sampling;

- Approximation-based attacks: algorithms based on this approach, such as BPDA, and EOT (see Table 2), respectively, make use of a differentiable function that approximates the outputs produced by a non-differentiable or a randomized layer of either a model or defense to feed traditional gradient-based attacks. The gradient-based attacks can then use the approximated outputs to perform the evasion [171].

3.3 Algorithms for Generating Adversarial Images

In Computer Vision, the algorithms used to generate adversarial perturbations are optimization methods that usually explore generalization flaws in pretrained models to craft and insert perturbations into legitimate images. The next sections will describe with more details four attack algorithms frequently used, namely, (i) FGSM [62], (ii) BIM [92], (iii) DeepFool [124], and (iv) CW
3.3.1 Fast Gradient Sign Method (FGSM). FGSM is a one-step algorithm proposed by Goodfellow et al. [62] to sustain his linear hypothesis for explaining the existence of adversarial examples (see Section 5.2). The main characteristic of FGSM is its low computational cost, resulted from perturbing, in just one step (limited by a given upper bound $\epsilon$), a legitimate image at the direction of the gradient that maximizes the model error. Despite its efficiency, the perturbations generated by FGSM are usually greater and less effective to fool models than the perturbations generated by iterative algorithms. Given an input image $x$, FGSM generates an adversarial image $x'$ according to Equation (2):
\[
x' = x - \epsilon \cdot \text{sign} \left( \nabla_x J(\Theta_f, x, y) \right),
\]
where $\epsilon$ is a hyperparameter that controls the magnitude of the perturbation, $\nabla_x$ represents the gradient vector, $\Theta_f$ represents the parameters of the model $f$, and $J(\Theta_f, x, y)$ the cost function used to train $f$.

3.3.2 Basic Iterative Method (BIM). This attack is an iterative version of FGSM, initially proposed by Kurakin et al. [92]. In contrast to FGSM, BIM executes several minor steps $\alpha$, where the total size of the perturbation is limited by an upper bound defined by the attacker. Formally, BIM can be defined as a recursive method, which generates $x'$ according to Equation (3):
\[
x' = \begin{cases} 
0, & \text{if } x_0 = 0, \\
\alpha \cdot \text{sign} \left( \nabla_x J(\Theta_f, x_{i-1}, y) \right), & \text{else}
\end{cases}
\]
where $\text{clip}$ limits values to the lower and higher edges outside the given interval.
perturb this image to make it cross the boundary and fool the classifier. Basically, DeepFool approximates, for each iteration, the solution of this problem by linearizing the classifier around an intermediate $x'$. The intermediate $x'$ is then updated toward the direction of an optimal direction by a small step $\alpha$. This process is repeated until the small perturbation computed by DeepFool makes $x'$ cross the decision boundary. Similar to FGSM, DeepFool is also based on the linearity hypothesis to craft perturbations.

3.3.4 Carlini and Wagner Attack (CW). The CW attack was proposed by Carlini and Wagner [22] and currently represents the state-of-the-art algorithm for generating adversarial images. Formally, given a DNN $f$ having a logits layer $z_f$ and a input image $x$ belonging to a class $y$, CW uses the gradient descent to solve iteratively the Equation (4):

$$\text{minimize } ||x - x'||^2_2 + c \cdot \ell(x'),$$

where, for $x$, the attack seeks for a small perturbation $\delta_x = x - x'$ that is able to fool the classifier. To do so, a hyperparameter $c$ is used as an attempt to compute the minimal amount of perturbation required. Besides $c$, there is the cost function $\ell(x')$, which is defined according to Equation (5):

$$\ell(x') = \max \{\max \{z_f(x')_i : i \neq y\} - z_f(x')_y, -conf\}. \quad (5)$$

In Equation (5), the hyperparameter $conf$ refers to the attack confidence rate. Higher $conf$ values contribute to generate adversarial images capable of fooling models with a high confidence rate, i.e., with predictions reaching probabilities up to 100% in a incorrect class $y' \neq y$. However, higher $conf$ values also produce adversarial images usually containing larger perturbations that are easily perceptible by humans.

Table 3 uses the introduced taxonomy to provide a comparative analysis of adversarial attacks by pointing out their advantages and disadvantages, in addition to listing some examples of application scenarios.

4 DEFENSES AGAINST ADVERSARIAL ATTACKS

The menace of adversarial images encouraged the scientific community to elaborate several approaches to defending classification models. However, designing such countermeasures is a difficult task once adversarial inputs are solutions to an optimization problem that is non-linear and non-convex. Since good theoretical tools for describing the solutions to these optimization problems do not exist, it is very hard to put forward a theoretical argument ensuring a defense strategy will be efficient against adversarial examples [91]. Therefore, the existing defense mechanisms have some limitations in the sense that they can provide robustness against attacks in specific threat models. The design of a robust machine learning model against all types of adversarial images and other examples is still an open research problem [25, p. 27].

4.1 Taxonomy of Defenses Against Adversarial Attacks

This section categorizes the defenses against adversarial attacks using a novel taxonomy, as depicted by Figure 4, that is composed of two different axes, namely, (i) defense objective and (ii) defense approach. As in Figures 2 and 3, all the axes introduced by the taxonomy in Figure 4 are in bold font for comparative purposes.

4.1.1 Defense Objective. According to its main objective, a defense can be (i) proactive or (ii) reactive. Proactive defenses aim to make classification models more robust to adversarial images. A model is considered robust when it can correctly classify an adversarial image as if it were a legitimate image. However, reactive defenses focus on detecting adversarial images by acting as
### Table 3. Comparative Analysis and Examples of Application Scenarios for Adversarial Attacks

| Axis | Type | Advantages | Disadvantages | Examples of Application Scenarios |
|------|------|------------|---------------|----------------------------------|
| **Attacker’s Influence** | Causative/Poisoning | (i) Evasive attacks, one-step cheap computation. | In real-world scenarios, it can be difficult for the attacker to access the original training dataset. | Online Machine Learning [41]. |
| | Evasive/Exploratory | There is a myriad of different algorithms and approaches available in the literature to attack and/or get information from models when they are in the inference stage. | There is plenty of research in Adversarial Machine Learning focused on protecting models from evasive attacks. | Integrity and privacy violations. |
| **Attacker’s Knowledge** | White-box Attacks | (i) Represent the most powerful type of attacks, given the fact that no defense so far was able to provide compelling protection. (ii) They are frequently used as a way to benchmark and validate the level of protection some defenses supposedly claim to offer. | It can be impracticable in real-world scenarios, since they require the attacker to have complete knowledge about the model’s parameters, architecture, and even existing defenses. | (i) Static evaluation of defenses. (ii) Generation of adversarial examples by using surrogate models. |
| | Black-box Attacks | They best simulate real-world scenarios, since the attacker has little or even no knowledge about the target model. | Black-box attacks tend to be the least evasive, and most of them can be simply neutralized by constraining the number of queries to the target model. | Attacks on deployed/MLaaS models. |
| | Grey-box Attacks | Intermediate attacks that evaluate defenses under harsher conditions than black-box scenarios, but constraining the attacker’s knowledge to only access the model’s parameters. | Grey-box attacks actually do not make much sense to simulate real-world scenarios. | A less strict alternative to assess models and defenses, especially when they present poor performance on white-box attacks [110]. |
| **Security Violation** | Integrity Violation | A stealth violation that degrades the performance of a model on the inference stage by using adversarial examples. It can result in harmful effects until be noticed. | In real-world scenarios, integrity violations rely most on black-box attacks. | Evasive attacks. |
| | Availability Violation | It makes the model completely unusable, causing a denial of service. | It depends on making the model’s performance degrading on legitimate samples. | Poisoning attacks. |
| | Privacy Violation | Values/labels of confidential information regarding the model and its training data are available. | With proper protective constraints, privacy violations can be very limited. | (i) Black-box attacks, (ii) Model inversion, (iii) Membership inference. |
| **Attack Specificity** | Targeted | It allows the attacker to control the specific output the model will misclassify. | (i) Targeted attacks can have more difficulties in finding high-confident examples [111]. (ii) Targeted adversarial examples are usually harder to transfer to other models [97]. | (i) Evasive attacks, (ii) Poisoning attacks. |
| | Untargeted | Untargeted adversarial examples are more easily found, usually with high confidence in the prediction. | The attacker cannot choose which class the model will misclassify on. | (i) Evasive attacks, (ii) Poisoning attacks. |
| **Attack Computation** | One-step | It usually crafts adversarial examples with coarser perturbations and with less confidence in the predictions. | It usually results in misclassifications. | (i) Evasive attacks, (ii) Poisoning attacks. |
| | Iterative | It usually crafts small and fine-tuned perturbations that maximize the classification error. | Expensive computation. | (i) Evasive attacks, (ii) Poisoning attacks. |
| **Attack Approach** | Gradient-based | Straightforward and usually a powerful approach that relies on the model’s gradients. | They are drastically affected by simply masking the model’s gradients or by adopting non-differentiable techniques. | (i) White-box attack, (ii) Black-box attack (through transferability). |
| | Transfer/Score-based | In contrast to gradient-based attacks, transfer/score-based attacks depend on the training data or the probabilities given by the target model. | In the training scenario, these attacks can be neutralized by varying the target model. | (i) Black-box attack. (ii) Attacks on gradient-masking defenses [171]. |
| | Decision-based | (i) It makes only on the model’s predicted class for a given example. (ii) It can be used along with other approaches such as gradient-based attacks. | (i) Usually needs many more iterations to converge [18]. (ii) It generally demands more queries to the target model to optimize the perturbations [18]. | (i) Black-box attack (ii) Attacks on gradient-masking defenses [171]. |
| | Approximation-based | It will be hard for attacking models and/or defenses containing non-differentiable or randomized components. | It takes longer to converge due to the gradient approximation process [15]. | (i) Attacks on gradient-masking defenses. (ii) Attacks on randomized models defenses [111]. |

* A learning process that consists of training a model on the fly by feeding it with streaming data instances [55, p.16].

MLaaS—Machine Learning as a Service.
a filter that identifies malicious images before they reach the classifier. The detected images are usually either discarded or sent to a recovery procedure.

4.1.2 Defense Approach. Defenses can adopt different approaches when protecting models against adversarial images. Each approach groups a set of similar procedures, which can range from brute force solutions to preprocessing techniques. Based on a systematic review of literature, this article also categorizes the most relevant proactive and reactive countermeasures according to their operational approach, which can be: (i) gradient masking, (ii) auxiliary detection models, (iii) statistical methods, (iv) preprocessing techniques. (v) ensemble of classifiers, and (vi) proximity measurements.

- Gradient Masking: defenses based on gradient masking (effect also known as obfuscated gradient [5]) produce, sometimes unintentionally, models containing smoother gradients that hinders optimization-based attack algorithms from finding wrong directions in space, i.e., without useful gradients for generating adversarial examples. According to Athalye et al. [5], defenses based on gradient masking can be organized in: (i) shattered gradients, (ii) stochastic gradients, and (iii) exploding/vanishing gradients.
  i. **Shattered gradients**: are caused by non-differentiable defenses, thus introducing nonexistent or incorrect gradients;
  ii. **Stochastic gradients**: are caused by randomized proactive/reactive defenses or randomized preprocessing on inputs before being fed to the classifier. This strategy of gradient masking usually leads an adversarial attack to incorrectly estimate the true gradient;
  iii. **Exploding/vanishing gradients**: are caused by defenses formed by very deep architectures, usually consisting of multiple iterations of a neural network evaluation, where the output of one layer is fed as an input of the next layer.

There are many countermeasures based on different strategies of gradient masking, as can be seen in Table 4. However, two distinct strategies are frequently mentioned by related works in literature, which, in turn, make them relevant to describe in more details: (i) Adversarial Training and (ii) Defensive Distillation.

Defenses based on adversarial training are usually considered in literature a brute force approach to protecting against adversarial examples. Essentially, the main objective of adversarial training is to make a classification model more robust by training it in a dataset containing legitimate and adversarial images. Formally, given a tuple $X = (x, y)$ and, for didactic purposes, $T$ a training dataset having only the tuple $X$, such as $T = \{X\}$, an adversarial image $x'$ is crafted from $x$ by an attack algorithm $A$, thus forming a new tuple $X'$ that will be assigned the same label $y$ of the clean image $x$, such that $X' = (x', y)$ and $x' = A(x)$. Afterwards, the training dataset $T$ is augmented with $X'$ and now contains two image tuples: $T = \{X, X'\}$. The learning model is then
Fig. 5. Adversarial training increases the robustness of classifiers by training them using an augmented training dataset containing adversarial images. Adapted from Shen et al. [152].

retrained using the augmented training dataset $T$, resulting in a theoretically stronger model (see Figure 5).

Despite the good results adversarial training presented in several works [62, 79, 85, 93, 117, 166, 172, 197], this gradient masking approach has two issues. The first issue is related to the strong coupling adversarial training has with the attack algorithm used during the training process. Retraining a model with adversarial training does not produce a generic model that is able to resist against evasions of adversarial images generated by a different attack algorithm not used in the training process. To have a more generic model, it would be necessary to elaborate a training dataset $T$ with a massive amount of adversarial images generated using different attack algorithms and amounts of disturbance. Therefore, the second issue concerning adversarial training raises: this is a procedure computationally inefficient, given two reasons: (i) the great number of adversarial images that must be crafted from different attacks, which, in turn, does not guarantee robustness against adversarial images generated from more complex algorithms and (ii) after generating these malicious images, the model must train using a much larger dataset, which exponentially grows the training time. A robust defense method must be decoupled from any attack algorithm to increase its generalization. Notwithstanding the drawbacks, Madry et al. [117] proposed training on adversarial samples crafted using Projected Gradient Descent (PGD) attack, which is, by the time of this writing, the most promising defense present in literature, since it demonstrated robustness to various types of attacks in both white-box and black-box settings [183]. However, their method is not model-agnostic, and, due to computational complexities, it has not been yet tested on large-scale datasets such as ImageNet [125].

Defensive Distillation, in turn, is a proactive defense proposed by Papernot et al. [139]. This countermeasure is originally inspired by a technique based on the transfer of knowledge among learning models known as distillation [75]. In learning distillation, the knowledge acquired by a complex model after training is transferred to a smaller model. Similarly, defensive distillation uses the knowledge of a model, represented by probabilistic vectors obtained from a first training, to perform a second training of the original model, as an attempt to smoothing its gradients and thus improving its robustness. Formally, defensive distillation begins by training a model $f$ using (i) a dataset $X$ containing the training samples, (ii) their corresponding labels $Y$, usually in one-hot encoding format, and (iii) a predefined temperature $t$ that is chosen beforehand by the defender. After training, the model $f$ provides the probabilistic vectors set $f(X)$ as output. The one-hot label set $Y$ is then replaced by the probabilistic vectors set $f(X)$, and a model $fd$ with the same architecture of $f$ is then created and trained with the sample set $X$, but now using as label set the probabilistic set $f(X)$. By the end of the training, the distilled probabilistic output $fd(X)$ is finally produced. Figure 6 depicts the schematic model of defensive distillation.
Defenses based on gradient masking usually produce models containing smoother gradients in certain regions of space, making them harder for the attacker to find promising directions to perturb an image. However, the attacker can instead use a non-differentiable attack, such as BPDA [5] or SPSA [176], as well as perform a black-box attack by training a surrogate model. This surrogate model reproduces the behavior of the targeted model, once the attacker queries it using images carefully crafted by him and watches the outputs the targeted model gives. Then, the attacker takes advantage of the transferability property of adversarial examples by using the gradients of the surrogate model to craft the images that will also lead the target model to misclassifications [60]. Section 5.7 gives more information regarding the transferability property of adversarial examples.

- **Auxiliary Detection Models (ADMs):** a defense based on ADMs is usually a reactive method that makes use of adversarial training to elaborate an auxiliary binary model that will act as a filter after being trained, checking whether an input image is legitimate or adversarial before sending it to the application classifier $f$. Works such as Gong et al., Grosse et al., Metzen et al., and Chen et al. [27, 59, 64, 120] proposed defenses based on ADMs.

Grosse et al. [64] adapted an application classifier $f$ to also act as an ADM, training it in a dataset containing $n + 1$ classes. The procedure followed by the authors consists of generating adversarial images $x'_i$ for each legitimate image $(x_i, y_i)$ that belongs to the training set $T$, where $i \leq |T| \times m$ (where $m$ is the number of attack algorithms used) and $j \leq n$. After the generation of adversarial images, it was formed a new training set $T_1$, where $T_1 = T \cup \{(x'_i, n + 1), i \leq |T| \times m\}$. $n + 1$ is the label assigned to an adversarial image. Finally, the model $f$ was trained using the $T_1$ set.

Gong et al. [59] elaborated a defense similar to Grosse et al. [64], but instead of adapting the application classifier to predict adversarial images in a class $n + 1$, the authors built and trained an ADM to filter out adversarial images $X'$ (crafted by FGSM and JSMA attacks) from the legitimate images $X$, using a training dataset $T_1$, formed from $T$. Formally, $T_1 = \{(x_i, 1) : i \in |T|\} \cup \{(x'_i, 0) : i \leq |T| \times m\}$.

In Metzen et al. [120], the representation outputs of the hidden layers of a DNN were used for training some ADMs, in a way similar to what was made in Reference [59]. The authors named these ADMs *subnetworks* and fixed them among specific hidden layers of a DNN to detect adversarial images. The authors performed the experiments using the FGSM, BIM, and DeepFool attack algorithms.

Finally, Chen et al. [27] elaborated a detection and reforming architecture called *ReabsNet*. When Reabsnet receives an image $x$, it uses an ADM (represented by a DNN trained by adversarial training) to check whether $x$ is legitimate or adversarial. In case of being classified as legitimate by the ADM, Reabsnet sends $x$ to the application classifier. However, in case of being classified as adversarial, $x$ is sent by ReabsNet to an iterative process, that reforms the image $x$ while it is classified as adversarial by the ADM. At the end of the reform process, the image $x$ is finally sent to the application classifier.
• **Statistical Methods:** Some works such as Grosse et al. and Feinman et al. [51, 64] performed statistical comparisons among the distributions of legitimate and adversarial images. Grosse et al. elaborated a reactive defense method that performs an approximation for the hypothesis test MMD (**Maximum Mean Discrepancy**) with the Fisher’s permutation test. The performed test verified if a legitimate dataset \( X_1 \) belonged to the same distribution of another dataset \( X_2 \), which might contain adversarial images. Formally, given two datasets \( X_1 \) and \( X_2 \), it is initially defined \( a = \text{MMD}(X_1, X_2) \). Later, there is a permutation of elements of \( X_1 \) and \( X_2 \) in two new datasets \( X'_1 \) and \( X'_2 \), and it is defined \( b = \text{MMD}(X'_1, X'_2) \). If \( a < b \), then the null hypothesis is rejected and then it is concluded that the two datasets belong to different distributions. This process is repeated several times and the **p-value** is defined as the fraction of the number of times that the null hypothesis was rejected.

Feinman et al. [51] also proposed a reactive defense called **Kernel Density Estimation (KDE)**. KDE makes use of Gaussian Mixture Models\(^2\) to analyze the outputs of the logits layer of a DNN and to verify if the input images belong to the same distribution of legitimate images. Given an image \( x \) classified as a label \( y \), the KDE estimates the probability of \( x \) according to Equation (6):

\[
\text{KDE}(x) = \frac{1}{|X_y|} \sum_{s \in X_y} \exp \left( \frac{(z_f(x) - z_f(s))^2}{\sigma^2} \right),
\]  

(6)

where \( X_y \) corresponds to the training samples that belong to class \( y \), \( z_f(x) \) is the logits output to input \( x \), \( z_f(s) \) is the logits output to a training sample \( s \), and \( \sigma \) is the kernel bandwidth. Therefore, the detector is built by the selection of a threshold \( \tau \), which classifies \( x \) as adversarial if \( \text{KDE}(x) < \tau \) or legitimate, otherwise.

• **Preprocessing Techniques:** other works elaborated countermeasures based on preprocessing techniques, such as image transformations [67, 187], GANs [146, 152], noise layers [108], autoencoders [66], and dimensionality reduction [74, 98, 190]. In the following, each work will be explained in more details.

Xie et al. [187] elaborated a proactive defense called **Random Resizing and Padding (RRP)** that inserts a resizing and a padding layer at the beginning of a DNN architecture. The resizing layer alters the dimensions of an input image, and later, the padding layer inserts null values in random positions on the surroundings of the resized image. At the end of the padding procedure, the resized image is classified by the proactive model.

Guo et al. [67] applied various transformations in input images before classification, such as cropping and rescaling, bit-depth reduction, JPEG compression, total variance minimization (TVM), and image quilting. Guo et al. implemented TVM as a defense by randomly picking pixels from the input and performing iterative optimization to find an image whose colors are consistent with the randomly picked pixels. However, image quilting involves reconstructing an image using small patches taken from the training database by using the nearest neighbor procedure (e.g., kNN\(^3\)). The intuition behind image quilting is to construct an image that is free from adversarial perturbations, since quilting only uses clean patches to reconstruct the image [183]. The authors claimed that TVM and image quilting presented the best results when protecting the classifier given the fact both (i) introduce randomness, (ii) are non-differentiable operations that hinder the attacker.

---

\(^2\)Gaussian Mixture Models are unsupervised learning models that clusters data by representing sub-populations, using normal distributions, within a general population.

\(^3\)kNN stands for k-Nearest Neighbors. It is a supervised classification algorithm that assigns, to a given input \( x \), the most frequent class among the \( k \) nearest training samples from \( x \), according to a certain distance metric.
to compute the model gradient, and (iii) are model-agnostic, which means the model does not need to be retrained or fine-tuned.

Shen et al. [152] proposed a proactive defense method that adapted a GAN to preprocess input images before they were sent to the application classifier. Samangouei et al. [146] also elaborated a defense based on a GAN framework that uses a generative transformation network $G$, which projects an input image $x$ onto the range of the generator by minimizing the reconstruction error $|G(\hat{x}) - x|^2$. After the transformation, the classifier is fed with the reconstruction $G(\hat{x})$. Since the generator was trained to model the unperturbed training data distribution, the authors claimed this added step results in a substantial reduction of any potential adversarial noise. In turn, Liu et al. [108] adopted an approach based on noise layers. These noise layers were inserted among the hidden layers of a CNN to apply a gaussian noise randomly crafted on each vector of the input image. According to the authors, this procedure avoids gradient-based attacks. Gu and Rigazio [66] elaborated the Deep Contractive Networks (DCNs), which are proactive defense methods that make use of denoising autoencoders and evolutionary algorithms as alternatives to remove perturbations from adversarial images.

In addition, some countermeasures preprocess input images using dimensionality reduction techniques [74, 98, 190]. These works are based on the hypothesis that, by reducing the dimensions of input, the likelihood of an attacker creating a perturbation that can affect the classifier’s performance decreases, given the fact the attack algorithm will have less information concerning the hyperspace of the image [190]. Keeping this hypothesis in mind, Hendrycks and Gimpel [74] designed a reactive defense based on Principal Component Analysis (PCA). The authors inferred that adversarial images assign greater weights on larger principal components and smaller weights on initial principal components. Li and Li [98] applied PCA on values produced by the convolution layers of a DNN and then used a cascade classifier to detect adversarial images. The cascade classifier $C$ classifies an image $x$ as legitimate only if all its subclasses $C_i$ classify $x$ as legitimate, but rejects $x$ if some classifier $C_i$ reject $x$. In this work, the L-BFGS attack was used to perform the experiments.

Xu et al. [190] introduced Feature Squeezing, which is a reactive defense that makes use of two techniques to reduce de dimensionality of an input image: (i) color bit depth reduction and (ii) spatial smoothing (or blur). The color bit depth reduction relies on the fact that all information regarding an image’s colors are actually irrelevant to the classification task, and, for this reason, they will not compromise the accuracy of the model when reduced. However, spatial smoothing stands for two groups of simple blurring techniques often used in image processing, namely, local smoothing and non-local smoothing, that are focusing on mitigating image noise. Both color bit depth reduction and spatial smoothing were chosen by the authors, since they are widely known, easy to implement, and computationally cheap. During the detection process, Feature Squeezing generates two reduced versions of an input image $x$: (i) $\hat{x}_1$, which represents image $x$ with the color bits reduced, and (ii) $\hat{x}_2$, which represents $x$ reduced with some spatial smoothing technique. Later, Feature Squeezing sends the images $x, \hat{x}_1 \in \hat{x}_2$ to be classified by a DNN $f$ and compares the softmax outputs $f(x), f(\hat{x}_1), f(\hat{x}_2)$ using the $L_1$ metric. If the $L_1$ metric exceeds a predefined threshold $\tau$, which is defined as a trade-off between high detection rates and false-positive rates, then Feature Squeezing classifies $x$ as an adversarial example and discards it. Figure 7 depicts this workflow.

- **Ensemble of Classifiers:** defenses based on an ensemble of classifiers are countermeasures formed by two or more classification models that can be chosen in runtime. This approach

---

4PCA is a dimensionality reduction technique that reduces, by applying a linear transformation, a set of points in $n$-dimensional space to a $k$-dimensional space, where $k \leq n$. 

ACM Computing Surveys, Vol. 55, No. 1, Article 8. Publication date: November 2021.
is based on the assumption that each model reciprocally compensates the weaknesses other models eventually might have when classifying a given input image [72]. Works such as Abbasi and Gagné, Strauss et al., Tramèr et al., and Sengupta et al. [1, 150, 162, 172] adopted different techniques to elaborate defenses based on ensemble of classifiers. Sengupta et al. [150] used a Bayesian algorithm to choose an optimal model from an ensemble to minimize the chances of evasion and, at the same time, maximize the correct predictions on legitimate images. Abbasi and Gagné [1] formed ensembles of specialist models that detects and classifies an input image by majority vote. Strauss et al. [162] made empirical evaluations based on four types of different ensembles and trainings. Tramèr et al. [172], in turn, used a variation of adversarial training to train the main classifier with adversarial images crafted by an ensemble of DNNs.

- **Proximity Measurements:** other works such as Cao and Gong, Carrara et al., Machado et al., Meng and Chen, Papernot and McDaniel proposed defenses based on proximity measurements among legitimate and adversarial images to the decision boundary. Papernot and McDaniel [135] elaborated a proactive defense method called **Deep k-Nearest Neighbors (DkNN),** which makes use of a variation of the kNN algorithm to compute uncertainty and reliability metrics from the proximity among the hidden representations of training and input images, obtained from each layer of a DNN. The labels representing points in space of training images are analyzed after the input image goes through all the layers of the DNN. In case of the prediction related to the input \(x\), given by the DNN, be in accordance with the labels representing the nearest training images to \(x\), the uncertainty metric tends to be small. In contrast, in the case of the labels of the training images be divergent among them, the uncertainty metric tends to be large [132]. Figure 8 depicts the DkNN operation.

Cao and Gong [17] also adopted an approach based on proximity metrics to elaborate a proactive countermeasure, called **Region-based Classification (RC).** RC is a variation of the kNN that defines a region \(R\) in hyperspace, having as centroid an input image \(x\), assigning it a label corresponding to the class that most intersects the area of this region \(R\). Formally, for a given input image \(x\) and a DNN \(f\) that splits the hyperspace in \(c\) distinct regions \(R = \{R_1, R_2, \ldots, R_c\}\) (\(c\) being the number of classes and \(R_i\) the predicted class for \(f(x)\)), it is created a hypercube \(B(x, r)\) around \(x\) (\(x\) being the centroid of \(B(x, r)\)) with length \(r\). \(Ar_i(B(x, r))\) is the area of hypercube \(B(x, r)\) that intersects the region \(R_i\). The classifier \(RC\) formed from \(f\) is defined as \(RC_{f,r}\) and its prediction for \(x\) is based on the region \(R_i\), which has the largest intersection with the area of hypercube, namely, \(RC_{f,r} = \arg\max_i(Ar_i(B(x, r)))\).

Carrara et al. [24] introduced a reactive defense that resembles somewhat the DkNN architecture [135]. The method proposed by the authors make use of a DNN \(f\), trained beforehand on a dataset \(T\), to classify a given unknown image \(x\). Afterwards, the activations obtained for \(x\) from a hidden layer of \(f\), chosen empirically, are used by a kNN algorithm to query \(T\) to get \(k\) images that have the most similar representations to \(x\). Therefore, a confidence score of \(s(x, f(x))\) is obtained, which

![Fig. 7. The Feature Squeezing workflow. Adapted from Reference [190].](image-url)
supports the prediction \( f(x) \), where such a score is computed based on similar samples returned by the kNN algorithm. If the confidence score is below a certain threshold \( \tau \), then the input image is classified as adversarial and discarded afterwards; otherwise, the prediction \( f(x) \) is said reliable. More formally, a model \( f \), pretrained on a dataset \( T = (X, Y) \), is used to predict a label \( f(x) \) for an given input image \( x \). Next, a kNN search is performed over \( T \) that returns a set \( \mathcal{N}_{\phi}^{f}(x, k) \) containing the most similar images to \( x \), such that \( \mathcal{N}_{\phi}^{f}(x, k) = \{(x_1, y_1), (x_2, y_2) \ldots (x_k, y_k)\} \), where \( \phi \) is the hidden layer of \( f \) chosen empirically, and \( L_2 \) is the euclidean distance \( d(x, x_i) = ||\phi(x) - \phi(x_i)||_2 \), with \( 1 \leq i \leq k, x_i \in X \), and \( y_i \in Y \). Then, a confidence score \( s(x, f(x)) \) is computed for each class \( y_i \) present in \( \mathcal{N}_{\phi}^{f} \) according to Equation (7):

\[
s(x, f(x)) = \frac{\sum_{i=1}^{k} w_i \cdot \mathbb{1}(f(x) = y_i)}{\sum_{i=1}^{k} w_i},
\]

where \( \mathbb{1}(f(x) = y_i) \) is equal to 1 when \( f(x) = y_i \), and 0 otherwise. In turn, \( w_i \) is a weight assigned to the \( i \)th result regarding the variant of the kNN algorithm used, such that \( w_i \in \{1, \frac{1}{d(x, x_i)}, \frac{1}{d(x, x_i)^2}\} \). The threshold \( \tau \) is chosen as a trade-off between false positives (i.e., the rate of adversarial examples whose scores are lesser than \( \tau \)) and true positives (i.e., the rate of legitimate examples whose scores are greater than \( \tau \)). After performing several experiments, the authors report the proposed method could detect, on average, 80% of adversarial images generated by the L-BFGS and FGSM attacks, in addition to classifying correctly more than 90% of the legitimate images.

In turn, Meng and Chen [119] proposed MagNet: a non-deterministic and reactive architecture composed of two defense layers that basically rely on autoencoders: (i) a detection layer, and (ii) a reformer layer. During the training on a dataset \( X \) containing only legitimate images, each autoencoder \( s_i \in \mathcal{S} \) (where \( i \leq n \), and \( n \) is the number of available autoencoders) is used to compute a subset \( M'_i \subset M' \) based on a predefined metric, which can be either the Reconstruction Error (RE) metric or the Jensen-Shannon Divergence (JSD) metric, such that \( M'_i = \{m_1, m_2, \ldots, m_k\} \), where \( k = |X|, r \in \{RE, JSD\} \). Afterwards, each \( M'_i \) set is sorted in descending order and a false positive rate \( 0 < \phi < 1 \), also defined beforehand, is applied to compute a index \( j = \lfloor k \times \phi \rfloor \), that is used to retrieve the chosen threshold \( \tau_i \) corresponding to the autoencoder \( s_i \), such that \( \tau_i = M'_i[j] \). When deployed, MagNet chooses randomly an autoencoder \( s_i \in \mathcal{S} \) when receives an input image \( x \), reconstructs it according to the chosen metric \( r \), such that \( \hat{x} = s_i(x) \), and computes a metric value \( m = r(x, \hat{x}) \) that is compared to the threshold \( \tau_i \). If \( m > \tau_i \), then the...
image \( x \) is considered adversarial and discarded; otherwise, it is sent to the reformer layer. The reformer layer acts as second line of defense, where is randomly chosen another autoencoder that reconstructs \( x \) and finally sent its reconstructed version to the application classifier. According to the authors, both layers complement each other, given the fact the detection layer rejects adversarial input images containing large perturbations whereas the reformer layer tries to remove any existing perturbations that might still be present in the filtered samples.

Based on the hypothesis that investing in randomness can produce better defenses [177], Machado et al. [116] extended the non-deterministic effect of the detection layer of MagNet by proposing a defense called MultiMagNet, which randomly chooses multiple defense components at runtime instead of just one, how is originally made by MagNet. In a way similar to MagNet, the MultiMagNet’s defense components were also implemented as autoencoders and trained on legitimate images. Later, the authors split the MultiMagNet’s architecture into two stages, namely, (i) calibration stage (which is usually executed just once) and (ii) deployment stage. During the calibration stage, MultiMagNet receives a set \( S \) containing \( n \) different autoencoders, and a file \( C \) containing three sets of hyperparameters, such that \( C = \Phi \cup M \cup A \). In file \( C \), \( \Phi \) represents the false positive rate set \( \phi \in \Phi, 0 < \phi < 1 \), \( M \) represents the metric set \( M = \{RE, JSD\} \), and \( A \) represents the threshold approach set, which can be either MTA (Multiple Threshold Approach) or minTA (Minimum Threshold Approach), thus producing \( c = |\Phi| \times |M| \times |A| \) combinations. All the \( c \) combinations are then evaluated for each autoencoder \( s_i \in S, i \leq n \) on a validation dataset containing both legitimate and adversarial images to find (i) the best threshold set \( T_b = \{\tau_1, \tau_2, \ldots, \tau_n\} \) that produces the highest accuracy, along with (ii) the combination of hyperparameters \( c_b \in C, c_b = \{\phi_b, m_b, a_b\} \), and \( \phi_b \in \Phi, m_b \in M, a_b \in A \). Both sets \( c_b \) and \( T_b \) will be used in the deployment stage. Once in deployment stage, MultiMagNet receives the input images to analyze before sending them to the application classifier. For each input image \( x \), MultiMagNet first forms randomly an ensemble \( S_x \subset S \) containing \( \sigma \) autoencoders, where \( 1 \leq \sigma \leq |S_x| \) and \( \sigma \mod 2 = 1 \), so as not to have ties. After forming \( S_x \), MultiMagNet loads from \( c_b \) the hyperparameter \( m_e \) to compute the metric set \( M_x \). Then, the thresholds corresponding to the chosen autoencoders in \( S_x \) are loaded in the set \( T_x \subset T \) according to hyperparameter \( a_b \). When \( a_b = MTA, \tau_i \in T_x = \{\tau_1, \tau_2, \ldots, \tau_n\} \). However, when \( a_b = minTA, \tau_i = min\{T_x\} \). Finally, the process of majority vote begins by comparing each metric \( m_i \in M_x \) to \( \tau_i \): in case of \( m_i \leq \tau_i, q_{leg} \) is incremented, otherwise \( q_{adv} \) is incremented. If \( q_{leg} < q_{adv} \), then the image \( x \) is rejected; otherwise, image \( x \) is classified as legitimate by the detector layer of MultiMagNet and goes to the reformer (as is done in MagNet). At last, the reformer image \( x \) is sent to the application classifier. The authors also conducted a comparative study with MagNet using legitimate and adversarial images crafted by FGSM, BIM DeepFool, and CW attacks, and concluded the increasing of the non-deterministic effect by choosing multiple components can lead to better defense architectures.

Table 4 makes a comprehensive overview of some relevant defenses against adversarial attacks in Computer Vision available in the literature, following the taxonomy presented in Section 4.1. It also sorts the defenses by their names/references in alphabetical order, in addition to showing which of them have already been circumvented by mentioning the corresponding works on adversarial attacks. Furthermore, Table 5 also adopts the introduced taxonomy of adversarial defenses to bring a comparative analysis of some advantages and disadvantages concerning each type of defense, in addition to mentioning possible application scenarios.

5 EXPLANATIONS FOR THE EXISTENCE OF ADVERSARIAL EXAMPLES

Developing an understanding about the existence and the properties of adversarial examples, by reasoning why they affect the prediction of machine learning models, is usually the first step taken into consideration when elaborating attacks and defenses in Adversarial Machine Learning [115].
The vulnerability that CNNs and other machine learning algorithms present before the malicious effects of adversarial attacks is popularly known as the Clever Hans Effect, a term somewhat popularized by the advent of CleverHans library [133]. This effect is named after a German horse called Hans. His owner used to claim Hans owned intellectual abilities by answering arithmetic questions that people made to it by tapping its hoof the number of times corresponding to the correct answer. However, after several experiments conducted on Hans, psychologists concluded the horse was not actually solving arithmetic questions, but instead it somehow developed the ability to identify behavioral signals made by the crowd, such as clappings and yellings, that warned it to stop hitting

ACM Computing Surveys, Vol. 55, No. 1, Article 8. Publication date: November 2021.
The table below provides a comparative analysis and examples of application scenarios for adversarial defenses.

| Defense Objective | Type                        | Advantages                                                                 | Disadvantages                                                                 | Examples of Application Scenarios                      |
|-------------------|-----------------------------|----------------------------------------------------------------------------|-------------------------------------------------------------------------------|--------------------------------------------------------|
| Proactive         | (i) They aim to produce models more robust to adversarial examples without the need for additional mechanisms. (ii) They generally perform well when protecting against black-box attacks [178]. | Most proactive defenses can be bypassed by using high-confidence adversarial examples [178]. | Protecting against black-box attacks.                 |
| Reactive          | Most reactive defenses are model-agnostic, and therefore they can be easily incorporated into working architectures. (i) They are usually more vulnerable to evasions in white-box settings than proactive defenses [20, 106]. (ii) Most reactive defenses simply discard the examples considered as adversarial. | Deployed/MLaaS models.                                                       |                                                      |

| Defense Approach  |                                |                                                                            |                                                                              |                                                        |
|-------------------|                                |                                                                            |                                                                              |                                                        |
| Gradient Masking  | They often present good performance when protecting against gradient-based attacks. | Gradient-free attacks, such as decision-based and approximation-based, can easily bypass these defenses. | Protecting against gradient-based attacks.               |
| Auxiliary Detection Models (ADM) | Binary defenses that can be easily attached to the application model's pipeline. | They strongly depend on Adversarial Training, and therefore present poor generalization to other attack algorithms not seen during training. | Deployed/MLaaS models.                                  |
| Statistical Methods | In general, they are computationally cheap and can spot adversarial images with larger distortions by comparing them with the distribution of legitimate images. | They showed so far to be inaccurate against adversarial examples containing optimal perturbations crafted by stronger attacks [20]. | Protecting against non-optimal adversarial examples. |
| Preprocessing Techniques | (i) They also tend to be computationally cheap. (ii) They can eliminate non-optimal perturbations. | On their own, they do not present enough robustness in white-box settings or against stronger attacks. | (i) Non-optimal adversarial examples. (ii) Black-box attacks. (iii) May support the design of more complex defenses. |
| Ensemble of Classifiers | An ensemble of classifiers can provide diversity and compensate for the weaknesses that each specific model has. | Naive ensembles can severely suffer from the transferability property of adversarial examples. | They may be adopted to support the design of randomized defenses [116]. |
| Proximity Measurements | They are sensitive to detect non-optimal perturbations, given the fact that they rely on spatial measurements among legitimate and adversarial samples. | They can have difficulties distinguishing between optimal adversarial and legitimate examples located close to the model's decision boundary. | (i) Protecting against non-optimal adversarial examples. (ii) They may support the design of more complex defenses. |

It is important to note that Hans did not develop an adaptive intelligence, but rather a means of perceiving and interpreting its surroundings to correctly answer the questions. Similar to Hans, learning models are usually able to give correct answers to complex problems, such as image recognition and classification, but without really learning from training data, which make them susceptible to adversarial attacks [56, 91]. Despite the absence of a unanimous agreement, the development of techniques to handle these adversarial examples is crucial for the advancement of AI systems.

ACM Computing Surveys, Vol. 55, No. 1, Article 8. Publication date: November 2021.
accepted explanation for the adversarial paradox,\(^5\) this section will describe some common hypotheses present in literature regarding the existence of adversarial images.

5.1 **High Non-linearity Hypothesis**

Szegedy et al. [166] first concerned about the existence of adversarial examples. The authors argued that adversarial examples exist due to the high non-linearity of deep neural networks, which contributes to the formation of low probability pockets in the data manifold that is hard to reach by sampling an input space around a given example (see Figure 9(a)). According to Gu and Rigazio and Song et al. [66, 159], the emergence of such pockets is given chiefly due to some deficiencies of objective functions, training procedures, and datasets limited in size and diversity of training samples, thus leading models to poor generalizations.

5.2 **Linearity Hypothesis**

Goodfellow et al. [62] contradicted the non-linearity hypothesis of Szegedy et al. by assuming DNNs have a very linear behavior caused by several activation functions like ReLU and sigmoid that perpetuates small perturbed inputs in the same wrong direction. As an attempt to underlie their explanation, the authors elaborated on the FGSM attack. Fawzi et al. [50] said that the robustness of a classifier is independent of the training procedure used and the distance between two classes is larger in high-order classifiers than in linear ones, suggesting that it is harder to find adversarial examples in deeper models. This explanation also goes against the non-linearity hypothesis of Szegedy et al. However, in contrast to the linearity hypothesis, Tabacof and Valle [167] found evidence the phenomenon of adversarial paradox may be a more complex problem, since results obtained from empirical experiments suggested that shallow classifiers present a greater susceptibility to adversarial examples than deeper models. Despite some works that criticize the linearity hypothesis, some relevant attacks (such as FGSM [62] and DeepFool [124]) and defenses (such as Thermometer Encoding [16]) were based on it.

5.3 **Boundary Tilting Hypothesis**

Tanay and Griffin [168], however, rejected the linear hypothesis proposed by Goodfellow et al. by assuming it is “insufficient” and “unconvincing.” They proposed instead a boundary tilting perspective to explain the adversarial paradox. This assumption, according to the authors, is more related to the explanation given by Szegedy et al., where a learned class boundary lies close to the training samples manifold, but this learned boundary is “tilted” with respect to this training manifold. Thereby, adversarial images can be generated by perturbing legitimate samples toward the classification boundary until they cross it. The amount of required perturbation is smaller as the tilting degree decreases, producing high-confidence and misleading adversarial examples, containing visually imperceptible perturbations. The authors also believe this effect might be a result of an overfitted model. Figure 9(b) shows a simplified illustration of the boundary tilting perspective compared with the Szegedy et al. hypothesis.

5.4 **High-dimensional Manifold**

Gilmer et al. [58], in concordance with other works such as Mahloujifar et al., Shafahi et al. and Fawzi et al. [49, 118, 151], said that the phenomenon of adversarial examples is the result from the high-dimensional nature of the data manifold. To show evidence, Gilmer et al. created a synthetic dataset for better controlling their experiments and used it afterwards to train a model.

\(^5\)Tanay and Griffin [168] defined this paradox as the disparity between the high-performance classification of state-of-the-art deep learning models against their susceptibility to small perturbations that differ so close from one class to another.
Fig. 9. Comparison between the Szegedy et al. and Tanay and Griffin’s hypotheses [168]. (a) Szegedy et al.’s hypothesis lies on the assumption that an image space is densely filled with low probability adversarial pockets. Similarly, (b) Tanay and Griffin’s hypothesis indicates the existence of tilted boundaries what contributes to the emergence of adversarial examples.

After training it, the authors observed that inputs correctly classified by the model were close to nearby misclassified adversarial inputs, meaning that learning models are necessarily vulnerable to adversarial examples, independently of the training procedure used. At last, based on empirical results, Gilmer et al. also denied the assumption that states adversarial examples lie on a different distribution when compared to legitimate data [57, 119, 146, 159].

5.5 Lack of Enough Training Data
Schmidt et al. [149] claim learning models must generalize in a strong sense, i.e., with the help of robust optimization to achieve robustness. The authors observed the existence of adversarial examples is not necessarily a shortcoming of specific classification models, but an unavoidable consequence of working in a statistical setting. After gathering some empirical results, the authors concluded that, currently, there are no working approaches that attain adversarial robustness, mainly because existing datasets are not large enough to train strong classifiers.

5.6 Non-robust Features Hypothesis
Ilyas et al. [82] provided a different explanation based on the assumption the existence of adversarial perturbations does not necessarily indicate flaws regarding learning models or training procedures, but actually regarding images’ features. By taking into account the human’s perception, the authors split the features into (i) robust features, that lead models to correctly predict the true class even when they are adversarially perturbed, and (ii) non-robust features, which are features derived from patterns in the data distribution that are highly predictive yet brittle, incomprehensible to humans and more susceptible to be perturbed by an adversary. To underlie their assumption, the authors proposed constructing a novel dataset formed by images containing solely robust features that were filtered out from the original input images by using the logits layer of a trained DNN. Then, this dataset was used to train another DNN that was used to perform a comparative study. The results led the authors to find evidence that adversarial examples might arise as a result of the presence of non-robust features, which goes in the opposite direction of what it is commonly believed: that adversarial examples are not necessarily tied to the standard training framework. Their conclusion is somehow related to the Schmidt et al. [149] work.

5.7 Explanations for Adversarial Transferability
As briefly mentioned in Section 4.1.2, adversarial examples make heavy use of the transferability property to drastically affect the performance of learning models even in more realistic scenarios, where the attacker does not have access to much or any information regarding the target classifier, as is simulated by grey and black-box settings. Adversarial transferability can be formally defined as the property that some adversarial samples have to mislead not only a target model $f$
but also other models $f'$ even when their architectures greatly differ [137]. Papernot et al. [136] split adversarial transferability into two main categories: (i) *intra-technique* transferability, which occurs between two models that share a similar learning algorithm (e.g., DNNs) and are trained using the same dataset, however, initialized with different parameters (e.g., transferability between two DNN architectures, such as VGG-16 and ResNet-152); (ii) *cross-technique* transferability, that occurs between two models that, respectively, belong to different learning algorithms (e.g., DNN and SVM), where can even perform different learning tasks, such as image classification and object detection. According to Wiyatno et al. [183], understanding the transferability phenomenon is critical not only to explain the existence of adversarial examples but to create safer machine learning models.

Some assumptions arose in literature as an attempt to explain adversarial transferability. The linearity hypothesis assumed by Goodfellow et al. [62] suggests the direction of perturbations may be the crucial factor that allows the adversarial effect transfer among models, since the disturbances end up acquiring similar functions through training. Tramèr et al. [173], in turn, hypothesized if adversarial transferability is actually a consequence of the intersection between the adversarial subspace of two different models. By estimating the number of orthogonal adversarial directions using a technique called *Gradient Aligned Adversarial Subspace*, they found that the separating distance between the decision boundaries of two models was, on average, smaller than the distances between any inputs to the decision boundaries, even on adversarially trained models. This suggests their adversarial subspaces were overlapped. At last, they also concluded that transferability is inherent to models that preserve non-robust properties when trying to learn feature representations of the input space, what according to the authors is not a consequence of a lack of robustness, but an intrinsic property of the learning algorithms themselves. Their findings agreed with the works of Liu et al. [111] and Ilyas et al. [82].

6 PRINCIPLES FOR DESIGNING AND EVALUATING DEFENSES

Defending robustly against adversarial attacks is still an open question. Carlini et al. [18] assert defenses often claim robustness against adversarial examples without carrying out common security evaluations, which, in turn, contributes to the construction of brittle and limited architectures that are rapidly broken by novel and adaptive attacks. For this reason, the authors defined a basic set of principles and methodologies that should be followed by both defenders and reviewers to check whether a defense evaluation is thorough and follows currently accepted best practices. This is crucial to prevent researchers from taking deceitful statements and conclusions about their works. In the following, some basic and relevant principles are listed and briefly explained based on Carlini et al.’s guide for properly evaluating general defenses. For further orientations, consult Reference [18].

6.1 Define a Threat Model

Defenses should always define a threat model where it states to be robust against adversarial attacks. It is important the threat model be described in detail, preferably following the taxonomy defined in Section 3.2, so that reviewers and attackers can restrict their evaluations under the requirements the defense affirms to be secure. For instance, a certain defense claims robustness under a threat model formed by evasive attacks conducted in a white-box scenario, where adversarial examples are generated by gradient and approximation-based attacks, using $L_2$ norm, and perturbation size less than 1.5. Based on this information, fair attackers that are interested in this defense must follow exactly what is specified by this threat model when designing their attacks.
6.2 Simulate Adaptive Adversaries

A good evaluation must test the limits of defenses by simulating adaptive adversaries that make use of its threat model to elaborate strong attacks. All settings and attack scenarios that stand a chance to bypass the defense should be taken into consideration without exceptions. An evaluation conducted only based on non-adaptive adversaries is of very limited utility, since the results produced by the experiments do not bring reliable conclusions that support the defense’s claims and its robustness bounds. A good evaluation will not try to support or assist the defense’s claims but will try to break it under its threat model at all costs. Therefore, weak attack settings and algorithms, such as FGSM attack\(^6\) must not be solely used. It is worth mentioning there are some relevant libraries available online for helping researchers to perform evaluations by simulating adaptive adversaries, such as CleverHans [133], Adversarial Robustness Toolbox [130], Foolbox [142], DEEPSEC [103], and AdvBox [63].

6.3 Develop Provable Lower Bounds of Robustness

Most works make use of empirical and heuristic evaluations to assess the robustness of their defenses. However, provable approaches are preferred, since they provide, when the proof is correct, lower bounds of robustness, which ensure the performance of the evaluated defense will never fall below that level. Nevertheless, provable evaluations usually suffer from the lack of generalization, since they get attached to the network architecture and a specific set of adversarial examples \(X'\), crafted using a certain attack algorithm, that was used in the experiments. This evaluation does not give any proofs that extend for other adversarial examples \(x' \notin X'\), which makes the statement less powerful. Circumvent these problems when developing provable lower bounds is an active research path (see Section 7.2.1).

6.4 Perform Basic Sanity Tests

Sanity tests are important to identify anomalies and antagonistic results that can lead authors to make incorrect conclusions. Carlini et al. [18] listed some basic sanity tests that should be run to complement the experiments and support the results.

- **Report model accuracy on legitimate samples**: while the protection of learning models against adversarial examples is a relevant security issue, a significant decrease on legitimate data on behalf of increasing the robustness of the model might be unreasonable for scenarios where the probability of an actual adversarial attack is low and the cost of misclassification is not high. For reactive defenses, it is important to evaluate how the rejection of perturbed samples can affect the accuracy of the model on legitimate samples. An analysis of a Receiver Operating Characteristic (ROC) curve may be helpful to check how the choice of a threshold for rejecting adversarial inputs can decrease the model’s clean accuracy;

- **Iterative versus one-step attacks**: iterative attacks are usually more powerful than one-step attacks. If adversarial examples crafted by a one-step algorithm can affect classification models more than examples crafted by iterative ones, then it can indicate that the iterative attack is not properly calibrated;

- **Increase the perturbation budget**: attacks when allowed to produce larger amounts of distortion in images usually fool classifiers more often than attacks with smaller perturbation

\(^6\)FGSM originally was implemented to support the linearity hypothesis made in Goodfellow et al. [62]. For this and other reasons related to the attack configurations, such as its one-step execution when computing perturbations, this attack is considered weak and untrustworthy to fully test defenses, usually used only to run sanity tests (see Section 6.4).
budgets. Therefore, if the attack success rate decreases as the perturbation budget increases, this attack algorithm is likely flawed;

- **Try brute force attacks:** it can be an alternative in scenarios where the attacks do not succeed very often. By performing a random search attack within the defense’s threat model can help the attacker or reviewer to find adversarial examples that were not found by standard adversarial attacks, which indicates these algorithms must be somehow improved. Carlini et al. recommend starting this sanity test by sampling random points at larger distances from the legitimate input, limiting the search to strictly smaller distortions whenever an adversarial example is found.

- **White-box versus black-box attacks:** white-box attacks are generally more powerful than black-box attacks, since the attacker has complete access to the model and its parameters. For this reason, gradient-based attacks should, in principle, present better success rates. If gradient-based attacks have worse performance when compared to other attack approaches, then it can indicate the defense is somehow performing a kind of gradient masking and the gradient-based attack needs calibration.

- **Attack similar undefended models:** proactive and reactive defenses typically introduce a couple of modifications in the networks to increase their robustness. However, it can be worth trying to remove these security components out from the model and evaluate it under attacks without any protection. If the undefended model appears robust nevertheless, then it can infer that the defense itself is not actually protecting the model.

### 6.5 Releasing of Source Code

It is crucial that all source code regarding the experiments and pre-trained models, including even their hyperparameters, be available to the community through online repositories so that interested reviewers can reproduce the evaluations made by the original work and ensure their correctness.

### 7 DIRECTIONS OF FUTURE WORK

#### 7.1 Overview of the Coverage of the Field

It was noticed during the development of this survey that the overall research in Adversarial Machine Learning regarding image classification tasks is not evenly distributed. Instead, there are some very active areas where similar approaches are often proposed, whereas other areas are not well covered and end up suffering from a lack of works. Table 6 makes use of the taxonomies introduced by this survey to provide an overview of the research coverage in the field, by pointing out overstudied and understudied scenarios based on the number of relevant works in the literature (see Sections 3.1, 3.2, and 4.1). Section 7.2 gives additional information by listing some promising paths for future works that can be explored.

#### 7.2 Promising Research Paths

It can be said that adversarial defenses are still in their infancy, despite the impressive growth of published works in the last years. Numerous important questions are waiting for answers, especially those referring to how to defend robustly against adversarial examples. This opens some promising research paths that will be detailed in the following.

- **Development of Theoretical Lower Bounds of Robustness.** Most defenses are limited to empirical evaluations and do not claim robustness to unknown attacks [183]. Other works, in turn, devise theoretical robustness bounds that do not generalize to different attacks and threat
### Table 6. Overview of the Research in Adversarial Machine Learning for Image Classification

| Taxonomy | Axis | Overstudied Scenario(s) | Understudied Scenario(s) |
|----------|------|-------------------------|--------------------------|
| Adversarial Images | Perturbation Scope | Individual-scoped | Universal-scoped |
| Perturbation Visibility | Optimal, Noise | Visible, Physical |
| Perturbation Measurement | $L_2$, $L_{\infty}$ | $L_0$, $L_1$ |
| Adversarial Attacks and Attackers | Attacker’s Influence | Evasive/Exploratory | Causative/Poisoning |
| Attacker’s Knowledge | White-box, Black-box | Grey-box |
| Security Violation | Integrity | Availability, Privacy |
| Attack Specificity | Targeted, Untargeted | |
| Attack Computation | Iterative | One-step |
| Attack Approach | Gradient-based, Transfer/Score-based | Decision-based, Approximation-based |
| Defenses | Defense Objective | Proactive, Reactive | |
| Defense Approach | Gradient Masking, ADMs, Ensemble of Classifiers, Preprocessing Techniques, Proximity Measurements | Statistical Methods, Hybrid Architectures* |

*Indistinguishable and Fooling Images are not actually considered as relevant areas of research.

1 Hybrid Architectures were not referred to as belonging to the taxonomy of defenses, since it was found no works that fit in this research area. Section 7.2.6 suggests them as a path for future work instead.

models studied. A promising research path is the investigation of properties that can theoretically guarantee general lower bounds of robustness to adversarial attacks (see Section 6.3).

#### 7.2.2 Unanimously Explanations Concerning the Existence and Transferability of Adversarial Examples

As can be seen in Section 5, there are already some explanations for the existence and transferability of adversarial examples. However, none of them is universally accepted due to the lack of proof. Developing unanimously provable explanations for these phenomena is relevant for the field of Adversarial Machine Learning, since they will guide future defenses to focus on solving the actual flaw and help the community understand better the inner workings of deep learning models.

#### 7.2.3 Devising of Efficient Attack Algorithms

Crafting strong adversarial examples is computationally expensive even on vanilla datasets. Applications that count on small response times, such as the traffic signal recognition system of an autonomous vehicle, require efficiency from attack algorithms when intercepting and perturbing the inputs. Attacks in black-box environments also impose more difficulties to the attacker, since he usually has available only a limited number of queries to the oracle\(^7\) to generate the perturbations. Regarding defenses, a good evaluation also needs testing numerous attacks, which can be computationally infeasible depending on the algorithms and datasets used. Therefore, devising strong and efficient adversarial attacks is a relevant research path for both fields of Adversarial Machine Learning.

#### 7.2.4 Comparative Study with Prior Work

As previously mentioned, a large amount of adversarial defenses emerges in literature, however only few of them perform a comparative study with other methods. A well-conducted study comparing different security approaches could help to foment results, in addition to revealing promising architectures for specific threat models.

---

\(^7\) In black-box attacks, the term oracle often represents the target model the attacker wants to fool.
7.2.5 Studies on More Recent Architectures. There exist a plethora of studies that perform attacks and design defenses targeting well-known CNN architectures. Nonetheless, only a few were conducted toward novel architectures, such as the Binarized Neural Networks (BNNs) and Neural Ordinary Differential Equations (Neural ODEs). Recent experiments have pointed out that both BNNs and Neural ODEs can surpass conventional CNN models in robustness, without sacrificing the accuracy on clean examples [44, 53, 80, 191]. Despite the positive results, there are not fully accepted answers that explain this improvement in robustness, which opens some paths of future research in this direction. Studies in this field of research can also help to the design of more efficient and robust models, in addition to providing further explanations regarding the inner workings of neural networks. In contrast to Huang et al. [80], which observed evidence of gradient-masking when assessing their Neural ODE-based defense with the SPSA attack, all defenses investigated by this survey that are based on BNN approaches were only evaluated on gradient-based attacks. This lets another open question whether BNNs can also actually perform well in other different attack approaches, such as decision-based or approximation-based attacks.

7.2.6 Development of Hybrid Defenses. The development of hybrid defenses can also be an encouraging research path. A hybrid defense means an architecture formed by different countermeasures, which can be organized on individual processes called modules. Each module would be responsible for performing some security procedure according to the approach that represents it. On each module, a component (represented by a defense or preprocessing method) would be randomly picked from a repository when receiving the input image. For instance, a hybrid defense could consist of three modules: a (i) reactive module, which randomly chooses a reactive defense from a repository to detect adversarial images; a (ii) preprocessing module that randomly processes the detected adversarial images that came from the reactive module; a (iii) proactive module, which similarly chooses at random a proactive defense to finally classify the input image. To the best of the authors’ knowledge, there is no work in the literature that adopted a similar approach or study, which, in turn, makes this path open for future opportunities.

8 FINAL CONSIDERATIONS
Deep Learning models revealed to be susceptible to attacks of adversarial nature, despite shown impressive abilities when solving complex cognitive problems, especially tasks related to Computer Vision, such as image classification and recognition. This vulnerability severely menaces the application of these learning algorithms in safety-critical scenarios, which, in turn, may jeopardize the development of the field if this security issue persists in the future. The scientific community has been struggling to find alternatives to defend against adversarial attacks practically, since this problem was first spotted by the work of Szegedy et al. [166]. The numerous proposed defenses, albeit promising at first, demonstrate to be brittle and ineffective to stop strong and adaptive attacks though. This arms race between attacks and defenses makes the field of Adversarial Machine Learning fairly dynamic and active, where the emergence of novel defense approaches almost daily plays a role in becoming review papers quickly outdated.

Before this chaotic scenario, this article aimed to attend the interested readerships by elaborating a comprehensive survey that gathers the most relevant research on Adversarial Machine Learning. It covered topics from Machine Learning basics to adversarial examples and attacks, nevertheless with an emphasis on giving the readers a defender’s perspective. An extensive review of the literature has allowed recent and promising defenses, not yet mentioned by other works, to be studied and categorized following a novel taxonomy. Moreover, existing taxonomies to organize adversarial examples and attacks were updated to cover further approaches. Furthermore, it gathered existing explanations for the existence and transferability of adversarial examples, listed some

ACM Computing Surveys, Vol. 55, No. 1, Article 8. Publication date: November 2021.
common policies that should be considered by both defenders and reviewers when, respectively, designing and evaluating security methods for deep learning models, and provided some promising paths for future work. In summary, the main contributions of this work were the following:

- The update of some existing taxonomies to categorize different types of adversarial images and novel attack approaches that were raised in the literature;
- An exhaustive review and discussion of defenses against adversarial attacks that were categorized using a novel taxonomy;
- An overview of understudied and overstudied scenarios using the introduced taxonomies, in addition to the discussion of promising research paths for future works.

Securing against adversarial attacks is crucial for the future of several applications. Therefore, this article was elaborated to provide a detailed overview of the area to help researchers to devise better and stronger defenses. To the best of the authors’ knowledge, this is the most comprehensive survey focused on adversarial defenses available in the literature and it is hoped that this work can help the community to make Deep Learning models reaching their prime-time soon.

REFERENCES

[1] Mahdieh Abbasi and Christian Gagné. 2017. Robustness to adversarial examples through an ensemble of specialists. Retrieved from https://arXiv:1702.06856.
[2] Naveed Akhtar and Ajmal Mian. 2018. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access* 6 (2018), 14410–14430.
[3] Moustafa Alzantot, Yash Sharma, Supriyo Chakraborty, and Mani Srivastava. 2018. Genattack: Practical black-box attacks with gradient-free optimization. Retrieved from https://arXiv:1805.11090.
[4] Anish Athalye and Nicholas Carlini. 2018. On the robustness of the cvpr 2018 white-box adversarial example defenses. Retrieved from https://arXiv:1804.03286.
[5] Anish Athalye, Nicholas Carlini, and David Wagner. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. Retrieved from https://arXiv:1802.00420.
[6] Anish Athalye and Ilya Sutskever. 2017. Synthesizing robust adversarial examples. Retrieved from http://arxiv.org/abs/1707.07397.
[7] Yassine Bakhti, Sid Ahmed Fezza, Wassim Hamidouche, and Olivier Déforges. 2019. DDSA: A defense against adversarial attacks using deep denoising sparse autoencoder. *IEEE Access* 7 (2019), 160397–160407.
[8] Shumeet Baluja and Ian Fischer. 2017. Adversarial transformation networks: Learning to generate adversarial examples. Retrieved from https://arXiv:1703.09387.
[9] Marco Barreno, Blaine Nelson, Anthony D. Joseph, and J. D. Tygar. 2010. The security of machine learning. *Machine Learning* 81, 2 (2010), 121–148. https://doi.org/10.1007/s10994-010-5188-5
[10] Joseph Bethge, Christian Bartz, Haojin Yang, Ying Chen, and Christoph Meinel. 2021. MeliusNet: An improved network architecture for binary neural networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 1439–1448.
[11] Arjun Nitin Bhagoji, Daniel Cullina, and Prateek Mittal. 2017. Dimensionality reduction as a defense against evasion attacks on machine learning classifiers. Retrieved from https://arXiv:1704.02654.
[12] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. 2013. Evasion attacks against machine learning at test time. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 387–402.
[13] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang et al. 2016. End to end learning for self-driving cars. Retrieved from https://arXiv:1604.07316.
[14] Wieland Brendel, Jonas Rauber, and Matthias Bethge. 2017. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. Retrieved from https://arXiv:1712.04248.
[15] Tom B. Brown, Dandelion Mané, Aurko Roy, Martin Abadi, and Justin Gilmer. 2017. Adversarial patch. Retrieved from https://arXiv:1712.09665.
[16] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. 2018. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*.
[17] Xiaoyu Cao and Neil Zhenviqiang Gong. 2017. Mitigating evasion attacks to deep neural networks via region-based classification. In *Proceedings of the 33rd Annual Computer Security Applications Conference (ACSAC’17)*. ACM, New York, NY, 278–287. https://doi.org/10.1145/3134600.3134606
Adversarial Machine Learning in Image Classification

[18] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, and Aleksander Madry. 2019. On evaluating adversarial robustness. Retrieved from https://arXiv:1902.06705.

[19] Nicholas Carlini, Guy Katz, Clark W. Barrett, and David L. Dill. 2017. Ground-truth adversarial examples. Retrieved from http://arxiv.org/abs/1709.10207.

[20] Nicholas Carlini and David Wagner. 2017. Adversarial examples are not easily detected: Bypassing ten detection methods. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. ACM, 3–14.

[21] Nicholas Carlini and David Wagner. 2017. MagNet and “efficient defenses against adversarial attacks” are not robust to adversarial examples. Retrieved from https://arXiv:1711.08478.

[22] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In Proceedings of the IEEE Symposium on Security and Privacy (SP’17). IEEE, 39–57.

[23] N. Carlini and D. Wagner. 2018. Audio adversarial examples: Targeted attacks on speech-to-text. In Proceedings of the IEEE Security and Privacy Workshops (SPW’18). 1–7. https://doi.org/10.1109/SPW.2018.00009

[24] Fabio Carrara, Fabrizio Falchi, Roberto Caldelli, Giuseppe Amato, and Rudy Becarelli. 2018. Adversarial image detection in deep neural networks. Multimedia Tools Appl. 78, 3 (2018), 2815–2835. DOI: https://doi.org/10.1007/s11042-018-5853-4

[25] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Deebdeep Mukhopadhyay. 2018. Adversarial attacks and defenses: A survey. Retrieved from https://arXiv:1810.00069.

[26] Jianbo Chen and Michael I. Jordan. 2019. HopSkipJumpAttack: A query-efficient decision-based attack. Retrieved from http://arxiv.org/abs/1904.02144.

[27] Jiefeng Chen, Zihang Meng, Changtian Sun, Wei Tang, and Yinglun Zhu. 2017. ReabsNet: Detecting and revising adversarial examples. Retrieved from https://arXiv:1712.08250.

[28] Jinyin Chen, Mengmeng Su, Shijing Shen, Hui Xiong, and Haibin Zheng. 2019. POBA-GA: Perturbation optimized black-box adversarial attacks via genetic algorithm. Comput. Secur. 85 (2019), 89–106. https://doi.org/10.1016/j.cose.2019.04.014

[29] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. 2018. Ead: Elastic-net attacks to deep neural networks via adversarial examples. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence.

[30] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 2017. Zoa: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (ABSEC’17). https://doi.org/10.1145/3128572.3140448

[31] Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Polo Chau. 2018. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 52–68.

[32] Moustapha Cisse, Yossi Adi, Natalia Neverova, and Joseph Keshet. 2017. Houdini: Fooling deep structured prediction models. Retrieved from https://arXiv:1707.05373.

[33] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. 2017. Parseval networks: Improving robustness to adversarial examples. In Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 854–863.

[34] Gilad Cohen, Guillermo Sapiro, and Raja Giryes. 2019. Detecting adversarial samples using influence functions and nearest neighbors. Retrieved from https://arXiv:1909.06872.

[35] G. E. Dahl, J. W. Stokes, L. Deng, and D. Yu. 2013. Large-scale malware classification using random projections and neural networks. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. 3422–3426. https://doi.org/10.1109/ICASSP.2013.6638293

[36] G. E. Dahl, Dong Yu, Li Deng, and A. Acero. 2012. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. Trans. Audio Speech Lang. Proc. 20, 1 (2012), 30–42. https://doi.org/10.1109/TASL.2011.2134090

[37] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Li Chen, Michael E. Kounavis, and Duen Horng Chau. 2017. Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. Retrieved from https://arXiv:1705.02900.

[38] Li Deng. 2012. Three classes of deep learning architectures and their applications: A tutorial survey. APSIPA Trans. Signal Info. Process. 3 (2012), e2. DOI: 10.1017/atip.2015.9

[39] Guneet S. Dhillon, Kamyar Azizzadenesheli, Zachary C. Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. 2018. Stochastic activation pruning for robust adversarial defense. Retrieved from https://arXiv:1803.01442.

[40] Lieyun Ding, Weili Fang, Hanbin Luo, Peter E. D. Love, Botao Zhong, and Xi Ouyang. 2018. A deep hybrid learning model to detect unsafe behavior: Integrating convolution neural networks and long short-term memory. Autom. Constr. 86 (2018), 118–124.

[41] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. 2018. Boosting adversarial attacks with momentum. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 9185–9193.
[42] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. 2019. Evading defenses to transferable adversarial examples by translation-invariant attacks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 4312–4321.

[43] Abhimanyu Dubey, Laurens van der Maaten, Zeki Yalniz, Yixuan Li, and Dhruv Mahajan. 2019. Defense against adversarial images using web-scale nearest-neighbor search. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 8767–8776.

[44] Kirsty Duncan, Ekaterina Komendantskaya, Robert Stewart, and Michael Lones. 2020. Relative robustness of quantized neural networks against adversarial attacks. In Proceedings of the International Joint Conference on Neural Networks (IJCNN’20). IEEE, 1–8.

[45] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for text classification. Retrieved from https://arXiv:1712.06751.

[46] Logan Engstrom, Andrew Ilyas, and Anish Athalye. 2018. Evaluating and understanding the robustness of adversarial logit pairing. Retrieved from https://arXiv:1807.10272.

[47] Ivan Evtimov, Kevin Eykholt, Earlene Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. 2017. Robust physical-world attacks on deep learning models. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security. 1528–1540. Retrieved from http://arxiv.org/abs/1707.08945.

[48] Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. 2017. Robust physical-world attacks on deep learning models. Retrieved from https://arXiv:1707.08945.

[49] Alhussein Fawzi, Hamza Fawzi, and Omar Fawzi. 2018. Adversarial vulnerability for any classifier. In Advances in Neural Information Processing Systems. MIT Press, 1178–1187.

[50] Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2015. Fundamental limits on adversarial robustness. In Proceedings of the ICML Workshop on Deep Learning.

[51] Reuben Feinman, Ryan R. Curtin, Saurabh Shintre, and Andrew B. Gardner. 2017. Detecting adversarial samples from artifacts. Retrieved from http://arxiv.org/abs/1703.00410.

[52] Joachim Fzli, Sebastian Palacio, Joern Hees, Damian Borth, and Andreas Dengel. 2018. Adversarial defense based on structure-to-signal autoencoders. Retrieved from https://arxiv.org/abs/1803.07994.

[53] Angus Galloway, Graham W. Taylor, and Medhat Moussa. 2017. Attacking binarized neural networks. Retrieved from https://arXiv:1711.00449.

[54] Ji Gao, Beilun Wang, Zeming Lin, Weilin Xu, and Yanjun Qi. 2017. Deepcloak: Masking deep neural network models for robustness against adversarial samples. Retrieved from https://arXiv:1702.07663.

[55] Aurélien Géron. 2019. Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O’Reilly Media.

[56] Dave Gershgorn. 2016. Fooling the machine: The byzantine science of deceiving artificial intelligence. Popular Sci. 30 (Mar. 2016).

[57] Partha Ghosh, Arpan Losalka, and Michael J. Black. 2019. Resisting adversarial attacks using gaussian mixture variational autoencoders. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33. 541–548.

[58] Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S. Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. 2018. Adversarial spheres. Retrieved from https://arXiv:1801.02774.

[59] Zhizhao Gong, Wenlu Wang, and Wei-Shinn Ku. 2017. Adversarial and clean data are not twins. Retrieved from https://arXiv:1704.04960.

[60] Ian Goodfellow and Nicolas Papernot. 2017. Is attacking Machine Learning easier than defending it? Retrieved from http://www.cleverhans.io/security/privacy/ml/2017/02/15/why-attacking-machine-learning-is-easier-than-defending-it.html.

[61] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In Advances in Neural Information Processing Systems. MIT Press, 2672–2680.

[62] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In Proceedings of the International Conference on Learning Representations. Retrieved from http://arxiv.org/abs/1412.6572.

[63] Dou Goodman, Hao Xin, Wang Yang, Wu Yuesheng, Xiong Junfeng, and Zhang Huan. 2020. Advbox: A toolbox to generate adversarial images that fool neural networks. Retrieved from https://arXiv:2001.05574.

[64] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. 2017. On the (statistical) detection of adversarial examples. Retrieved from http://arxiv.org/abs/1702.06280.

[65] Kathrin Grosse, Nicolas Papernot, Praveen Manoharan, Michael Backes, and Patrick McDaniel. 2017. Adversarial examples for malware detection. In Proceedings of the European Symposium on Research in Computer Security (ESORICS’17), Simon N. Foley, Dieter Gollmann, and Einar Snekkens (Eds.). Springer International Publishing, Cham, 62–79.
[66] Shixiang Gu and Luca Rigazio. 2014. Towards deep neural network architectures robust to adversarial examples. Retrieved from http://arxiv.org/abs/1412.5068.
[67] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. 2017. Countering adversarial images using input transformations. Retrieved from https://arXiv:1711.00117.
[68] Jamie Hayes and George Danezis. 2018. Learning universal adversarial perturbations with generative models. In Proceedings of the IEEE Security and Privacy Workshops (SP’18). IEEE, 43–49.
[69] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR’16). IEEE, 770–778. https://doi.org/10.1109/CVPR.2016.90
[70] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 770–778.
[71] Warren He, Bo Li, and Dawn Song. 2018. Decision boundary analysis of adversarial examples. In International Conference on Learning Representations.
[72] Warren He, James Wei, Xinyun Chen, Nicholas Carlini, and Dawn Song. 2017. Adversarial example defenses: Ensembles of weak defenses are not strong. In Proceedings of the 11th USENIX Workshop on Offensive Technologies (WOOT’17). Retrieved from https://www.usenix.org/system/files/conference/woot17/woot17-paper-he.pdf.
[73] J. B. Heaton, N. G. Polson, and Jan Hendrik Witte. 2017. Deep learning for finance: Deep portfolios. Appl. Stochast. Models Bus. Industry 33, 1 (2017), 3–12.
[74] Dan Hendrycks and Kevin Gimpel. 2017. Early methods for detecting adversarial images. In Proceedings of the 5th International Conference on Learning Representations (ICLR’17).
[75] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. In Proceedings of the Deep Learning and Representation Learning Workshop at NIPS 2014.
[76] Geoffrey E. Hinton and Ruslan R. Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. Science 313, 5786 (2006), 504–507.
[77] Jie Hu, Li Shen, and Gang Sun. 2017. Squeeze-and-excitation networks. Retrieved from https://arXiv:1709.01507.
[78] J. B. Heaton, N. G. Polson, and Jan Hendrik Witte. 2017. Deep learning for finance: Deep portfolios. Appl. Stochast. Models Bus. Industry 33, 1 (2017), 3–12.
[79] Hyunseung Jung, Kang Hwa, and Hyung Ju Yoo. 2017. Rethinking adversarial training. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR’17). IEEE, 1–9.
[80] Hyunseung Jung, Kang Hwa, and Hyung Ju Yoo. 2017. Rethinking adversarial training. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR’17).
[81] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. 2019. Adversarial examples are not bugs, they are features. Retrieved from https://arXiv:1905.02175.
[82] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 1125–1134.
[83] Vishaal Munusamy Kabilan, Brandon Morris, and Anh Nguyen. 2018. VectorDefense: Vectorization as a defense to adversarial examples. Retrieved from https://arXiv:1804.08529.
[84] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. 2018. Adversarial logit pairing. Retrieved from https://arXiv:1803.06373.
[85] Danny Karmon, Shbani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. 2019. Adversarial examples are not bugs, they are features. Retrieved from https://arXiv:1905.02175.
[86] Andrej Karpathy. 2014. What I learned from competing against a ConvNet on ImageNet. Retrieved from http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet.
[87] Erica Klarreich. 2016. Learning securely. Commun. ACM 59, 11 (2016), 12–14.
[88] Erick Knorr. 2015. How paypal beats the bad guys with machine learning. InfoWorld (Apr. 2015). Retrieved from https://www.infoworld.com/article/2907877/machine-learning/how-paypal-reduces-fraud-with-machine-learning.html.
[89] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. Communications of the ACM 60, 6 (2012), 1097–1105.
[90] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. Retrieved from https://arXiv:1607.02533.
[91] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial machine learning at scale. Retrieved from https://arXiv:1611.01236.
[94] Alexey Kurakin, Ian Goodfellow, Samy Bengio, Yinpeng Dong, Fangzhou Liao, Ming Liang, Tianyu Pang, Jun Zhu, Xiaolin Hu, Cihanx Xie et al. 2018. Adversarial attacks and defences competition. In Proceedings of the NIPS’17 Competition: Building Intelligent Systems. Springer, 195–231.

[95] Alex Lamb, Jonathan Binas, Anirudh Goyal, Dmitriy Serdyuk, Sandeep Subramanian, Ioannis Mitliagkas, and Yoshua Bengio. 2019. Fortified Networks: Improving the Robustness of Deep Networks by Modeling the Manifold of Hidden Representations. Retrieved from https://openreview.net/forum?id=SkgVRic9Km.

[96] Changjiang Li, Haisin Weng, Shouling Ji, Jianfeng Dong, and Qinming He. 2019. DeT: Defending against adversarial examples via decreasing transferability. In Proceedings of the International Symposium on Cyberspace Safety and Security. Springer, 307–322.

[97] Maosen Li, Cheng Deng, Tengjiao Li, Junchi Yan, Xinbo Gao, and Heng Huang. 2020. Towards transferable targeted attack. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 641–649.

[98] Xin Li and Fuxin Li. 2016. Adversarial examples detection in deep networks with convolutional filter statistics. Retrieved from https://arXiv:1612.07767.

[99] Xin Li and Fuxin Li. 2016. Adversarial examples detection in deep networks with convolutional filter statistics. Retrieved from https://arXiv.org/abs/1612.07767.

[100] Bin Liang, Hongcheng Li, Miaofang Su, Xirong Li, Wenchang Shi, and Xiaofeng Wang. 2017. Detecting adversarial examples in deep networks with adaptive noise reduction. Retrieved from http://arxiv.org/abs/1705.08378.

[101] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. 2018. Defense against adversarial attacks using high-level representation guided denoiser. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 1778–1787.

[102] Ji Lin, Chuang Gan, and Song Han. 2019. Defensive quantization: When efficiency meets robustness. Retrieved from https://arXiv:1904.08444.

[103] Xiang Ling, Shouling Ji, Jiaxu Zou, Jiannan Wang, Chunming Wu, Bo Li, and Ting Wang. 2019. Deepsec: A uniform platform for security analysis of deep learning model. In Proceedings of the IEEE Symposium on Security and Privacy (SP’19). IEEE, 673–690.

[104] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I. Sánchez. 2017. A survey on deep learning in medical image analysis. Med. Image Anal. 42 (2017), 60–88.

[105] Ninghao Liu, Hongxia Yang, and Xia Hu. 2018. Adversarial detection with model interpretation. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 1803–1811.

[106] Qiang Liu, Pan Li, Wentao Zhao, Wei Cai, Shui Yu, and Victor C. M. Leung. 2018. A survey on security threats and defensive techniques of machine learning: A data driven view. IEEE Access 6 (2018), 12103–12117.

[107] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyi Zeng, Yurong Liu, and Fuad E. Alsaadi. 2017. A survey of deep neural network architectures and their applications. Neurocomputing 234 (Nov. 2016), 11–26. https://doi.org/10.1016/j.neucom.2016.12.038

[108] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. 2017. Towards robust neural networks via random self-ensemble. Retrieved from https://arXiv:1712.00673.

[109] Xuanqing Liu, Yao Li, Chongruo Wu, and Cho-Jui Hsieh. 2018. ADV-BNN: Improved adversarial defense through robust bayesian neural network. Retrieved from https://arxiv.org/abs/1810.01279.

[110] Xuanqing Liu, Si Si, Qin Cao, Sanjiv Kumar, and Cho-Jui Hsieh. 2019. Neural SDE: Stabilizing neural ode networks with stochastic noise. Retrieved from https://arXiv:1906.02355.

[111] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. 2016. Delving into transferable adversarial examples and black-box attacks. Retrieved from https://arXiv:1611.02770.

[112] Zihao Liu, Qi Liu, Tao Liu, Yanzhi Wang, and Wujie Wen. 2018. Feature distillation: DNN-Oriented JPEG compression against adversarial examples. Retrieved from https://arXiv:1803.05787.

[113] Jiajun Lu, Theerasit Issaranon, and David Forsyth. 2017. Safetynet: Detecting and rejecting adversarial examples robustly. In Proceedings of the IEEE International Conference on Computer Vision. 446–454.

[114] Shiqing Ma, Yingqi Liu, Guanhong Tao, Wen-Chuan Lee, and Xiangyu Zhang. 2019. NIC: Detecting adversarial samples with neural network invariant checking. In Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS’19).

[115] Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E. Houle, and James Bailey. 2018. Characterizing adversarial subspaces using local intrinsic dimensionality. Retrieved from https://arXiv:1801.02613.

[116] Gabriel R. Machado, Ronald R. Goldschmidt, and Eugénio Silva. 2019. MultiMagNet: A non-deterministic approach based on the formation of ensembles for defending against adversarial images. In Proceedings of the 21st International Conference on Enterprise Information Systems (ICEIS’19). INSTICC, SciTePress, 307–318. https://doi.org/10.5220/0007714203070318

ACM Computing Surveys, Vol. 55, No. 1, Article 8. Publication date: November 2021.
Adversarial Machine Learning in Image Classification

[117] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. Retrieved from https://arXiv:1706.06083.

[118] Saeed Mahloujifar, Dimitrios I. Diochnos, and Mohammad Mahmoody. 2019. The curse of concentration in robust learning: Evasion and poisoning attacks from concentration of measure. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33. 4536–4543.

[119] Dongyu Meng and Hao Chen. 2017. Magnet: A two-pronged defense against adversarial examples. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security. ACM, 135–147.

[120] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. 2017. On detecting adversarial perturbations. Retrieved from https://arXiv:1702.04267.

[121] Jan Hendrik Metzen, Mummadi Chaitanya Kumar, Thomas Brox, and Volker Fischer. 2017. Universal adversarial perturbations against semantic image segmentation. Stat 1050 (2017), 19.

[122] Takeru Miyato, Andrew M. Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. Retrieved from https://arXiv:1605.07725.

[123] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, Pascal Frossard, and Stefano Soatto. 2017. Analysis of universal adversarial perturbations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR’17). https://doi.org/10.1109/CVPR.2017.17 arXiv:1705.09554

[124] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: A simple and accurate method to fool deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2574–2582.

[125] Seyed-Mohsen Moosavi-Dezfooli, Ashish Shrivastava, and Oncel Tuzel. 2018. Divide, denoise, and defend against adversarial attacks. Retrieved from https://arXiv:1802.06806.

[126] Aamir Mustafa, Salman H. Khan, Munawar Hayat, Jianhing Shen, and Ling Shao. 2019. Image super-resolution as a defense against adversarial attacks. Retrieved from http://arxiv.org/abs/1901.01677.

[127] Taesik Na, Jong Hwan Ko, and Saibal Mukhopadhyay. 2017. Cascade adversarial machine learning regularized with a unified embedding. Retrieved from https://arXiv:1708.02582.

[128] Nina Narodytska and Shiva Kasiviswanathan. 2017. Simple black-box adversarial attacks on deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW’17), IEEE, 1310–1318.

[129] Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 427–436. https://doi.org/10.1109/CVPR.2015.7298640 arXiv:1412.1897

[130] Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Ambirsh Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, Ian Molloy, and Ben Edwards. 2018. Adversarial robustness toolbox v0.3.0. Retrieved from https://arXiv.org/pdf/1807.01069.

[131] Tianyu Pang, Chao Du, and Jun Zhu. 2017. Robust deep learning via reverse cross-entropy training and thresholding test. 3. Retrieved from https://arXiv:1706.00633.

[132] Nicolas Papernot. 2018. A marauder’s map of security and privacy in machine learning. Retrieved from https://arXiv:1811.01314.

[133] Nicolas Papernot, Fartash Faghri, Nicholas Carlini, Ian Goodfellow, Reuben Feinman, Alexey Kurakin, Chihang Xie, Yash Sharma, Tom Brown, Aurko Roy, Alexander Matyasko, Vahid Behzadan, Karen Hambardzumyan, Zhishuai Zhang, Yi-Lin Juang, Zhi Li, Ryan Sheatsley, Abhishek Garg, Jonathan Usato, Willi Gierke, Yinpeng Dong, David Berthelot, Paul Hendricks, Jonas Rauber, and Rujun Long. 2018. Cleverhans v2.1.0: An adversarial machine learning library. Retrieved from https://arXiv:1610.00768.

[134] Nicolas Papernot and Patrick McDaniel. 2017. Extending defensive distillation. Retrieved from https://arxiv.org/abs/1705.05264.

[135] Nicolas Papernot and Patrick McDaniel. 2018. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. Retrieved from https://arXiv:1803.04765.

[136] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. 2016. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. Retrieved from https://arXiv.org/pdf/1605.07277.pdf.

[137] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In Proceedings of the ACM Asia Conference on Computer and Communications Security (ASIA CCS’17). 506–519. https://doi.org/10.1145/3052973.3053009 arXiv:1602.02697

[138] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. 2016. The limitations of deep learning in adversarial settings. In Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P’16). IEEE, 372–387.

[139] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016. Distillation as a defense to adversarial perturbations against deep neural networks. In Proceedings of the IEEE Symposium on Security and Privacy (SP’16). 582–597. https://doi.org/10.1109/SP.2016.41 arXiv:1511.04508
[140] Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. 2018. Deflecting adversarial attacks with pixel deflection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 8571–8580.

[141] Rajeev Ranjan, Swami Sankaranarayanan, Carlos D Castillo, and Rama Chellappa. 2017. Improving network robustness against adversarial attacks with compact convolution. Retrieved from https://arXiv:1712.00699.

[142] Jonas Rauber, Wieland Brendel, and Matthias Bethge. 2017. Foolbox: A python toolbox to benchmark the robustness of machine learning models. Retrieved from http://arxiv.org/abs/1707.04131.

[143] Andras Rozsa, Ethan M. Rudd, and Terrance E. Boult. 2016. Adversarial diversity and hard positive generation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 25–32.

[144] Yibin Ruan and Jiazhu Dai. 2018. TwinNet: A double sub-network framework for detecting universal adversarial perturbations. Future Internet 10, 3 (2018), 26.

[145] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet large-scale visual recognition challenge. Int. J. Comput. Vision 115, 3 (2015), 211–252. https://doi.org/10.1007/s11263-015-0816-y

[146] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. 2018. Defense-gan: Protecting classifiers against adversarial attacks using generative models. Retrieved from https://arXiv:1805.06605.

[147] Sayantan Sarkar, Ankan Bansal, Upal Mahbub, and Rama Chellappa. 2017. UPSET and ANGRI: Breaking high performance image classifiers. Retrieved from https://arXiv:1707.01159.

[148] Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. Neural Netw. 61 (2015), 85–117.

[149] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. 2018. Adversarially robust generalization requires more data. In Advances in Neural Information Processing Systems. MIT Press, 5014–5026.

[150] Sailik Sengupta, Tathagata Chakraborti, and Subbarao Kambhampati. 2017. MTDeep: Securing deep neural nets against adversarial attacks with moving target defense. Retrieved from http://arxiv.org/abs/1705.07213.

[151] Ali Shafahi, W. Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. 2018. Are adversarial examples inevitable? Retrieved from https://arXiv:1809.02104.

[152] S. Shen, G. Jin, K. Gao, and Y. Zhang. 2017. APE-GAN: Adversarial perturbation elimination with GAN. Retrieved from https://arXiv:1707.05474.

[153] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In Proceedings of the IEEE Symposium on Security and Privacy (SP’17). IEEE, 3–18.

[154] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. 2018. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. Science 362, 6419 (2018), 1140–1144.

[155] Taylor Simons and Dah-Jye Lee. 2019. A review of binarized neural networks. Electronics 8, 6 (2019), 661.

[156] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. Retrieved from https://arXiv:1409.1556.

[157] Aman Sinha, Hongseok Namkoong, and John Duchi. 2017. Certifying some distributional robustness with principled adversarial training. Retrieved from https://arXiv:1710.10571.

[158] Chawin Sitawarin and David Wagner. 2019. On the robustness of deep k-nearest neighbors. Retrieved from https://arXiv:1903.08333.

[159] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. 2017. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. Retrieved from https://arXiv:1710.10766.

[160] Vignesh Srinivasan, Arturo Marban, Klaus-Robert Müller, Wojciech Samek, and Shinichi Nakajima. 2018. Counter-strike: Defending deep learning architectures against adversarial samples by langevin dynamics with supervised denoising autoencoder. Retrieved from https://arXiv:1805.12017.

[161] Shriansh Srivastava, J. Priyadarshini, Sachin Gopal, Sanchay Gupta, and Har Shobhit Dayal. 2019. Optical character recognition on bank cheques using 2D convolution neural network. In Applications of Artificial Intelligence Techniques in Engineering. Springer, 589–596.

[162] Thilo Strauss, Markus Hanselmann, Andrej Junginger, and Holger Ulmer. 2017. Ensemble methods as a defense to adversarial perturbations against deep neural networks. Retrieved from https://arXiv:1709.03423.

[163] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. 2019. One pixel attack for fooling deep neural networks. IEEE Trans. Evol. Comput. 23, 5 (2019), 828–841.

[164] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR’15). 1–9. DOI: 10.1109/CVPR.2015.7298594
Advances in Machine Learning in Image Classification

[165] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2818–2826. Retrieved from http://arxiv.org/abs/1512.00567.

[166] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. In Proceedings of the International Conference on Learning Representations. 1–10. Retrieved from http://arxiv.org/abs/1312.6199.

[167] Pedro Tabacof and Eduardo Valle. 2016. Exploring the space of adversarial images. In Proceedings of the International Joint Conference on Neural Networks (IJCNN’16). IEEE, 426–433.

[168] Thomas Tanay and Lewis Griffin. 2016. A boundary tilting perspective on the phenomenon of adversarial examples. Retrieved from https://arXiv:1608.07690.

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

[170] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, and Javier Ortega-Garcia. 2018. Exploring recurrent neural networks for on-line handwritten signature biometrics. IEEE Access 6, 5128–5138 (2018), 1–7.

[171] Florian Tramèr, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. 2020. On adaptive attacks to adversarial example defenses. Retrieved from https://arXiv:2002.08347.

[172] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Dan Boneh, and Patrick McDaniel. 2017. Ensemble adversarial training: Attacks and defenses. Retrieved from http://arxiv.org/abs/1705.07204.

[173] Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. 2017. The space of transferable adversarial examples. Retrieved from https://arXiv:1704.03453.

[174] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. 2016. Stealing machine learning models via prediction apis. In Proceedings of the 25th USENIX Security Symposium (USENIXSecurity’16). 601–618.

[175] Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. 2019. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33. 742–749.

[176] Jonathan Uesato, Brendan O’Donoghue, Aaron van den Oord, and Pushmeet Kohli. 2018. Adversarial risk and the dangers of evaluating against weak attacks. Retrieved from https://arXiv:1802.05666.

[177] Yevgeniy Vorobeychik and Murat Kantarcioglu. 2018. Adversarial machine learning. Synth. Lect. Artif. Intell. Mach. Learn. 12, 3 (2018), 1–169.

[178] Derui Wang, Chaoran Li, Sheng Wen, Surya Nepal, and Yang Xiang. 2020. Defending against adversarial attack towards deep neural networks via collaborative multi-task training. IEEE Trans. Depend. Secure Comput. (2020), 1–12. DOI: 10.1109/TDSC.2020.3014390

[179] Gang Wang, Tianyi Wang, Haitao Zheng, and Ben Y. Zhao. 2014. Man vs. machine: Practical adversarial detection of malicious crowdsourcing workers. In Proceedings of the 23rd USENIX Security Symposium (USENIXSecurity’14). 239–254.

[180] Jianyu Wang and Haichao Zhang. 2019. Bilateral adversarial training: Towards fast training of more robust models against adversarial attacks. In Proceedings of the IEEE International Conference on Computer Vision. 6629–6638.

[181] Yizhen Wang and Kamalika Chaudhuri. 2018. Data poisoning attacks against online learning. Retrieved from https://arXiv:1808.08994.

[182] Wenqi Wei, Ling Liu, Margaret Loper, Stacey Truex, Lei Yu, Mehmet Emre Gursoy, and Yanzhao Wu. 2018. Adversarial examples in deep learning: Characterization and divergence. Retrieved from https://arXiv:1807.00051.

[183] Rey Reza Wiyatno, Anqi Xu, Ousmane Dia, and Archy de Berker. 2019. Adversarial examples in modern machine learning: A review. Retrieved from https://arXiv:1911.05268.

[184] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. 2018. Generating adversarial examples with adversarial networks. Retrieved from https://arXiv:1801.02610.

[185] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. 2018. Spatially transformed adversarial examples. Retrieved from https://arxiv.org/abs/1801.02612.

[186] Huang Xiao. 2017. Adversarial and Secure Machine Learning. Ph.D. Dissertation. Universität München. Retrieved from https://mediatum.ub.tum.de/1335448.

[187] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. 2017. Mitigating adversarial effects through randomization. Retrieved from https://arXiv:1711.01991.

[188] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L. Yuille, and Kaiming He. 2019. Feature denoising for improving adversarial robustness. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR’19).

[189] Boyan Xu, Ruichu Cai, Zhenjie Zhang, Xiaoyan Yang, Zhifeng Hao, Zijian Li, and Zhihao Liang. 2019. NADAQ: Natural language database querying based on deep learning. IEEE Access 7 (2019), 35012–35017. DOI: 10.1109/ACCESS.2019.2904720

ACM Computing Surveys, Vol. 55, No. 1, Article 8. Publication date: November 2021.
[190] Weilin Xu, David Evans, and Yanjun Qi. 2018. Feature squeezing: Detecting adversarial examples in deep neural networks. In Proceedings of the Network and Distributed Systems Security Symposium (NDSS’18). https://doi.org/10.14722/ndss.2018.23198

[191] Hanshu Yan, Jiawei Du, Vincent Y. F. Tan, and Jiashi Feng. 2019. On robustness of neural ordinary differential equations. Retrieved from https://arXiv:1910.05513.

[192] Ziang Yan, Yiwen Guo, and Changshui Zhang. 2018. Deep Defense: Training DNNs with improved adversarial robustness. In Advances in Neural Information Processing Systems. MIT Press, 419–428.

[193] Yuzhe Yang, Guo Zhang, Dina Katabi, and Zhi Xu. 2019. ME-Net: Towards effective adversarial robustness with matrix estimation. Retrieved from https://arXiv:1905.11971.

[194] Ziqi Yang, Ee-Chien Chang, and Zhenkai Liang. 2019. Adversarial neural network inversion via auxiliary knowledge alignment. Retrieved from https://arXiv:1902.08552.

[195] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. 2019. Adversarial examples: Attacks and defenses for deep learning. IEEE Trans. Neural Netw. Learn. Syst. 30, 9 (2019), 2805–2824. DOI: 10.1109/TNNLS.2018.2886017

[196] X. Yuan, C. Li, and X. Li. 2017. DeepDefense: Identifying DDoS attack via deep learning. In Proceedings of the IEEE International Conference on Smart Computing (SMARTCOMP’17). 1–8. https://doi.org/10.1109/SPECIALCOMP.2017.7946998

[197] Valentina Zantedeschi, Maria-Irina Nicolae, and Ambrish Rawat. 2017. Efficient defenses against adversarial attacks. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. ACM, 39–49.

[198] Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In Proceedings of the European Conference on Computer Vision. Springer, 818–833.

[199] Zhihao Zheng and Pengyu Hong. 2018. Robust detection of adversarial attacks by modeling the intrinsic properties of deep neural networks. In Advances in Neural Information Processing Systems. MIT Press, 7913–7922.

[200] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. Retrieved from https://arXiv:1703.10593.

Received February 2020; revised June 2021; accepted August 2021