Adversarial Attacks and Defenses in Deep Learning: From a Perspective of Cybersecurity

SHUAI ZHOU, CHI LIU, DAYONG YE, and TIANQING ZHU,
University of Technology Sydney, Australia
WANLEI ZHOU, City University of Macau, Macau SAR, China
PHILIP S. YU, University of Illinois at Chicago, USA

The outstanding performance of deep neural networks has promoted deep learning applications in a broad set of domains. However, the potential risks caused by adversarial samples have hindered the large-scale deployment of deep learning. In these scenarios, adversarial perturbations, imperceptible to human eyes, significantly decrease the model's final performance. Many papers have been published on adversarial attacks and their countermeasures in the realm of deep learning. Most focus on evasion attacks, where the adversarial examples are found at test time, as opposed to poisoning attacks where poisoned data is inserted into the training data. Further, it is difficult to evaluate the real threat of adversarial attacks or the robustness of a deep learning model, as there are no standard evaluation methods. Hence, with this article, we review the literature to date. Additionally, we attempt to offer the first analysis framework for a systematic understanding of adversarial attacks. The framework is built from the perspective of cybersecurity to provide a lifecycle for adversarial attacks and defenses.

CCS Concepts: • **Theory of computation** → *Adversarial learning*; • **Security and privacy** → *Usability in security and privacy;*

Additional Key Words and Phrases: Deep learning, adversarial attacks and defenses, cybersecurity, advanced persistent threats

ACM Reference format:
Shuai Zhou, Chi Liu, Dayong Ye, Tianqing Zhu, Wanlei Zhou, and Philip S. Yu. 2022. Adversarial Attacks and Defenses in Deep Learning: From a Perspective of Cybersecurity. *ACM Comput. Surv.* 55, 8, Article 163 (December 2022), 39 pages.
https://doi.org/10.1145/3547330

1 INTRODUCTION

Machine learning techniques have been applied to a broad range of scenarios and have achieved widespread success, especially for deep learning, which is fast becoming a key instrument in

This article is supported by an ARC project, DP190100981 and DP200100946, from the Australian Research Council, Australia, and NSF under grants III-1763325, III-1909323, III-2106758, and SaTC-1930941.
Authors' addresses: S. Zhou, C. Liu, D. Ye, and T. Zhu (corresponding author), University of Technology Sydney, P.O. Box 123, Broadway NSW 2007, Australia; emails: {Shuai.zhou, Chi.liu}@student.uts.edu.au, {Dayong.ye, Tianqing.zhu}@uts.edu.au; W. Zhou, City University of Macau, P.O. Box 123, Avenida Padre Tomás Pereira Taipa, China; email: wlzhou@cityu.edu.mo; P. S. Yu, University of Illinois at Chicago, P.O. Box 123, Chicago, Illinois; email: psyu@uic.edu.

0360-0300/2022/12-ART163 $15.00
https://doi.org/10.1145/3547330

various tasks. However, in many scenarios, the failure of a machine learning or deep learning model can cause serious safety problems. For example, in autonomous vehicles, failure to recognize a traffic sign could lead to a severe accident [1]. Hence, it is critical to train an accurate and stable model before it is deployed on a large scale. Unfortunately, in recent years, many studies have revealed a disappointing phenomenon in model security in that deep learning models might be vulnerable to the adversarial examples, i.e., samples that have been perturbed by an adversary maliciously. With high probability, models that have been tampered with in this way will produce wrong predictions, even though they may show high accuracy with benign samples [2–5]. Adversarial attacks can then be broadly defined as a class of attacks that aim to fool a machine learning model by inserting adversarial examples into either the training phase, known as a poisoning attack [6–8], or the inference phase, called an evasion attack [2, 3]. Either attack will significantly decrease the robustness of the deep learning models and raise the model security problems. Moreover, the vulnerabilities of deep learning solutions beset with this model security problem have been recently uncovered in the real world, which has led to the concerns over how much we can trust deep learning technologies.

Due to the diversity of potential threats about privacy and security in the practical applications of deep learning techniques, more and more organizations, such as ISO, IEEE, and NIST, are participating in the process of standardizing artificial intelligence. For some countries, this undertaking is considered to be akin to the construction of new infrastructure in some countries [9]. The ISO has proposed a project concerning the lifecycle of AI systems, which divides the technology's lifecycle into eight stages, including initialization, design and development, inspection and verification, deployment, operation monitoring, continuous verification, re-evaluation, and abandonment [10]. What is not further addressed in this cycle is how adversarial attacks are hindering the commercial deployment of deep learning models. For this reason, evaluating the threats to model security is a critical component in the lifecycle of an AI project. And, further, given the fragmented, independent, and diverse nature of possible adversarial attacks and defenses, how a model's security threats are analyzed should also be standardized. What is urgently needed is a risk map to help accurately determine the multiple types of risks at each stage of a project's lifecycle. More seriously, the defense of the attacks is still in the early stage, so more sophisticated analysis technology is highly required.

Some surveys related to adversarial machine learning have been published in recent years. Chakraborty et al. [11] described the catastrophic consequences of adversarial examples in security-related environments and review some strong countermeasures. However, their conclusions showed that none of them can act as a panacea for all challenges. Hu et al. [12] first introduced the lifecycle of an AI-based system, which is used to analyze the security threats and research advances at each stage. Adversarial attacks are allocated into training and inference phases. Serban et al. [13] and Machado et al. [14] reviewed existing works about adversarial machine learning in object recognition and image classification, respectively, and summarized the reasons why adversarial examples exist. Serban et al. [13] also described the transferability of adversarial examples between different models. Similar to Reference [14] providing relevant guidance to devise the defenses, Zhang et al. [15] also provided a comprehensive survey on relevant works from the defender's perspective and the summarized hypotheses of the origin of adversarial examples for deep neural networks. There are also some surveys regarding the applications of adversarial machine learning to specific domains such as recommender systems [16], cybersecurity domain [17], and medical systems [18].

It has previously been observed that, in the cybersecurity context, **Advanced Persistent Threats (APT)** are usually highly organized and have an extremely high likelihood of success [19]. Take Stuxnet, for example—this is one of the most famous APT attacks. It was launched

in 2009 and took down Iran's nuclear weapon program [19]. The workflow of APT considers the security problems systematically, which allows APT-related technologies to both achieve outstanding success rate, i.e., to bypass those defenses, and to evaluate the security of the system with a high degree of efficacy. Inspired by the workflow of APT, we have applied this systematic analysis tool to cybersecurity problems as a potential way to analyze the threats of an adversarial attack.

APTs mainly consist of multiple types of existing underlying cyberspace attacks (such as SQL injection and malware). The combined strategies of different kinds of underlying attacks and their five-stage workflow mean APTs enjoy extremely high success rates compared to that of a single attack. Interestingly, however, it is possible to neatly fit the existing adversarial attacks into those five stages according to their attack strategies. Based on this observation, we find that a workflow similar to the APT works when evaluating the threat of adversarial attacks. Thus, this forms the basis of our analysis and countermeasures framework. Though some review papers have summarized works in model security, the attack methods or defenses are still generally classified into dependent and segmented classes. This means the relationships between different approaches have not been identified clearly. In this article, we provide a comprehensive and systematic review of the existing adversarial attacks and defenses systematically from the perspective of APT. Our contributions can be itemized as follows:

- We provide a novel cybersecurity perspective to investigate the security issues of deep learning. For the first time, we propose to incorporate the APT concept into the analysis of adversarial attacks and defenses in deep learning. The result is a standard APT-like analysis framework for model security. Unlike previous surveys conducted with a partial focus on, say, mechanisms [13, 20], threat models [21], or scenarios [22], our work can offer a global and system-level view for understanding and studying this problem. Specifically, previous studies tend to discuss the methods with similar strategies in groups. Adversarial attacks with different strategies are studied separately, which ignores the relationship between attacks falling into different groups. Instead, our work regards adversarial attacks as a global system, and each group of attacks with similar strategies is just a part of this global system. Similar to cybersecurity, considering the relationship between different groups can help boost the effectiveness of attacks further.
- Based on the APT-like analysis framework, we performed a systematic review regarding existing adversarial attack methods. In line with the logic of APT, adversarial attacks can be clearly classified into five stages. In each stage, the common essential components and short-term objectives are identified, which help to improve the attacking performance in a in-depth order.
- We also reviewed the defenses against adversarial attacks within the APT-like analysis framework. Likewise, defenses are divided into five stages, providing a top-down sequence to eliminate the threats of adversarial examples. Relationships between defensive methods at different stages can be identified, motivating a possible strategy of combining multiple defenses to provide higher robustness for deep learning models.
- We summarized the hypotheses for the existence of adversarial examples from the perspective of data and models, respectively, and provided a comprehensive introduction of commonly used datasets in adversarial machine learning.

We hope this work will inspire other researchers to view the model security risks (and even privacy threats) at the system level and to evaluate those risks globally. If a standard can be established, then the various properties, such as the robustness, could be accessed more accurately and in less time. As a result, confidence in deep learning models would increase for their users.

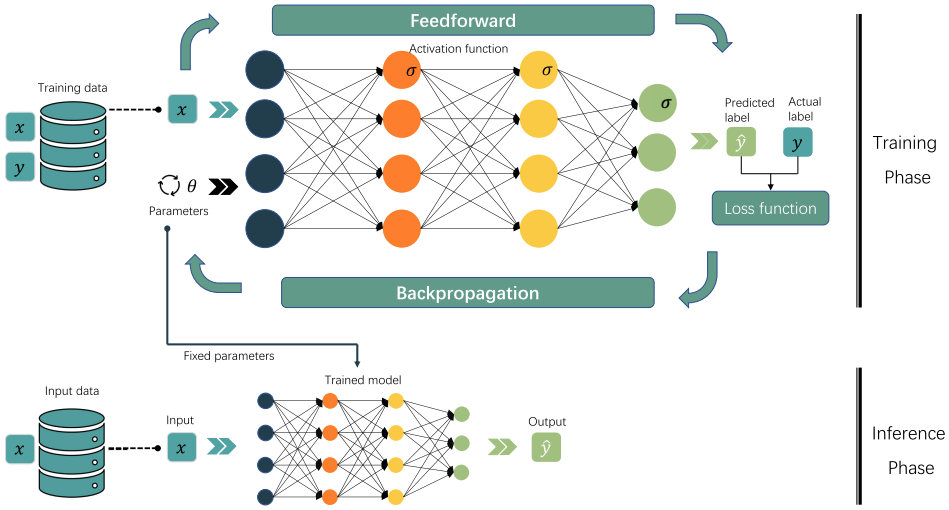

Fig. 1. The general workflow of a deep learning system. In the training phase, parameters θ are updated iteratively based on training data. After getting optimal parameters $\theta^\star$, input would be fed into the trained model in the inference phase, which will provide a corresponding output for decision.

2 PRELIMINARY

2.1 Deep Learning as a System

Deep learning refers to a set of machine learning algorithms built on **deep neural networks (DNNs)** and has been widely applied in tasks such as prediction and classification [21]. DNNs are a kind of mathematical model comprising multiple layers with a large number of computational neurons and nonlinear activation functions. The workflow of a typical deep learning system includes two phases: the training phase and the inference phase. The detailed processes of the two phases are shown as follows, as well as in Figure 1:

(1) In the training phase, the parameters of the DNN are updated continuously through iterative feedforwards and backpropagations. The gradient descending direction in backpropagation is guided by optimizing the loss function, which quantifies the error between the predicted label and the ground-truth label. Specifically, given a input space $\mathcal{X}$ and a label space $\mathcal{Y}$, the optimal parameters $\theta^\star$ of the DNN f are expected to minimize the loss function $\mathcal{L}$ on the training dataset $(\mathcal{X}, \mathcal{Y})$. Therefore, the training process to find the optimal $\theta^\star$ can be defined as:

$$\theta^\star = \arg\min_{\theta} \sum_{x_i \in \mathcal{X}, y_i \in \mathcal{Y}} \mathcal{L}(f_\theta(x_i), y_i),$$

where f_θ is the DNN model to be trained; $x_i \in \mathcal{X}$ is a data instance sampled from the training dataset, and y_i and $f_\theta(x_i)$ indicate the corresponding ground-truth label and the predicted label, respectively.

(2) In the inference phase, the trained models $f_\theta^\star$ with fixed optimal parameters $\theta^\star$ are applied to provide decisions on unseen inputs that are not included in the training dataset. Given an unseen input x_j, the corresponding model decision y_j (i.e., the predicted label of x_j) can be computed through a single feedforward process: $y_j = f_\theta^\star(x_j)$. It is worth noting that a successful adversarial attack often results in a manipulated prediction $\hat{y}$ in the inference phase, which could be far from the correct label of x_j.

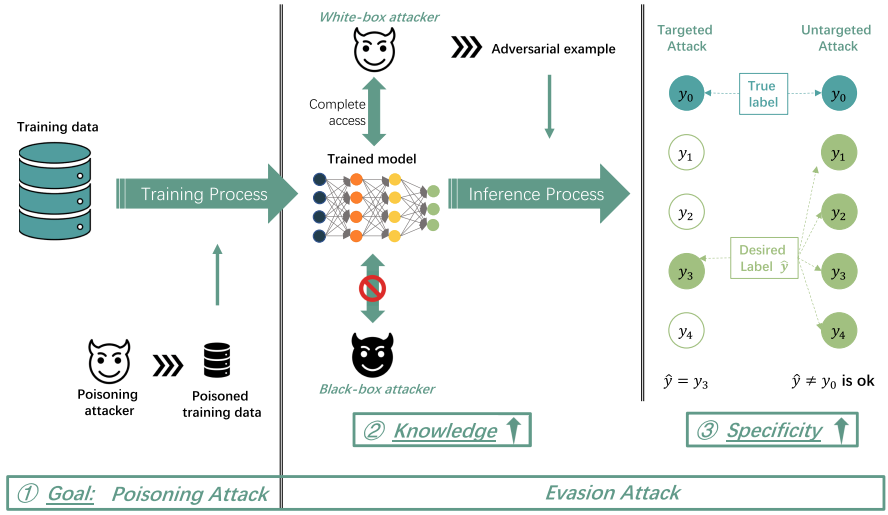

Fig. 2. Threat models in adversarial attacks, with illustrating the adversary's goal, the adversarial specificity and the adversary's knowledge. Poisoning attacks target the training phase (on the left side of the figure). The adversary's knowledge is also illustrated in the middle, while the right side shows an example result of adversarial specificity, where targeted attacks have more rigorous requirements for success, i.e., the desired output is a fixed label.

2.2 Threat Models against DNN

To classify these attacks, we have specified the threat model against DNN by introducing the critical components of the model attacks. The threat is decomposed into three dimensions: the adversary's goal, the adversarial specificity, and the adversary's knowledge. These three dimensions can help us identify potential risks and understand the attacking behaviors in the adversarial attack setting. In Figure 2, we provide an overview of the threat models in adversarial attacks.

2.2.1 *Adversary's Goal.*

- **Poisoning Attacks.** In poisoning attacks, the attackers can access and modify the training dataset to impact the final trained models [23–26]. The way that the attackers inject fake samples into the training data to generate a defective model can be viewed as "poisoning." Poisoning attacks generally lead to a decrease of accuracy [25] or misclassification on the given test samples [26].
- **Evasion Attacks.** In evasion attacks, the adversary's goal is to attack a well-trained and fixed DNN without any authority to modify the parameters of the target models [13, 18, 23]. In this way, the accessibility of the training dataset is no longer needed by the attackers. Instead, the attackers generate deceptive test samples that the target models fail to recognize to evade final detection [27, 28].

2.2.2 Adversarial Specificity. The difference in adversarial specificity depends on whether the attacker can predefine a specific fraudulent prediction for a given adversarial sample at the inference phase.

- **Untargeted Attacks.** In untargeted attacks, the adversary's only purpose is to fool the target model into generating a false prediction without caring for which label is chosen as the final output [29–32].

- **Targeted Attacks.** In targeted attacks, for a given sample, the attacker not only wants the target model to make an incorrect prediction, but also aims to induce the model to provide a specific false prediction [33–36]. Generally, targeted attacks do not succeed as often as an untargeted one.

2.2.3 Adversary's Knowledge.

- **White-box Attacks.** In white-box settings, the adversary is able to access the details of the target model, including structure information, gradients, and all possible parameters [30, 34, 37]. The adversary thus can craft elaborate adversarial samples by exploiting all the information at hand.
- **Black-box Attacks.** In black-box settings, attackers implement attacks without any knowledge of the target model. An attacker can acquire some information by interacting with the target model [36, 38, 39]. This is done by feeding input query samples into the model and analyzing the corresponding outputs.

2.3 Adversarial Attack

2.3.1 Evasion Attacks. Szegedy et al. [2] first introduced the concept of adversarial example in adversarial attacks, which can mislead the target model with a high success rate in the inference phase. They proposed a method to search for minimal distorted adversarial examples with the targeted label through Equation (1):

$$\text{minimize } \|x' - x\|_2^2 \text{ subject to } f(x') = t \text{ and } x' \in [0,1]^m. \tag{1}$$

Through this equation, they can find the closest x' that has a minimal distance with benign sample x by minimizing $\|x' - x\|_2^2$ and would be misclassified as targeted label t by the condition of $f(x') = t$. This problem can lead to the objective in Equation (2), which can be solved by L-BFGS algorithms:

$$\text{minimize } c\,\|x' - x\|_2^2 + \mathcal{L}(f(x'), t) \text{ subject to } x' \in [0,1]^m. \tag{2}$$

Evasion attacks can be loosely described as methods of crafting adversarial examples by adding imperceptible perturbations, which can result in the misbehavior of trained models.

2.3.2 Poisoning Attacks. Unlike evasion attacks happening in the inference attacks, poisoning attacks aim to downgrade the accuracy of models by polluting the training data. Attackers need some authorities to manipulate the training data, such as data injection and data modification [11]. As a result, the goals of launching poisoning attacks can be categorized into two classes: availability violation and integrity violation. The former aims to reduce the confidence or accuracy of victim model and disrupt the entire system, while the latter tries to mislead the victim model over some specific samples by introducing a backdoor without affecting other normal samples [11]. Specifically, the poisoned instances against neural networks only involving training samples can be crafted via following two strategies: bi-level optimization and feature collision [23].

- **Bi-level optimization:** Classical data poisoning of modifying the data can be formalized as a bi-level optimization problem [23]. However, for non-convex neural networks, bi-level optimization problems are intractable. Muñoz-González et al. [24] proposed "back-gradient descent" to approximate solutions of the inner problem and then conduct gradient descent on the outer loss, although it is still computationally expensive. To speed up the production of poisoned samples, Yang et al. [25] introduced GAN to generate poisons. MetaPoison et al. [26] is also proposed as a first-order method to approximately solve the bi-level optimization of producing poisons using the ensembling strategies.

- **Feature collision:** Methods based on bi-level optimization usually are effective against both transfer learning and end-to-end training, while feature collision strategy can be used to design efficient attacks against transfer learning in the targeted misclassification setting. For instance, Shafahi et al. [40] developed a method to generate the poisoning sample similar to the target samples in the feature space, while it is close to the original benign sample in the input space. These two categories of methods only require permission to manipulate the training instead of the label, and the semantic of training data will remain. These poisoned samples with clean-label will be more difficult to be detected.

Moreover, in addition to poisoning attacks only manipulating training data, another type of poisoning attacks are backdoor attacks, which need the additional capacity of inserting trigger to the input in inference phase [23].

- **Backdoor attacks:** Adversaries in backdoor attacks usually have access to modify the label of training samples [41]. These mislabeled data with backdoor triggers will be injected into training dataset. As a result, the trained model based on this dataset will be forced to assign the new sample (with the trigger) the desired target label. Most backdoor attacks require mislabeling the training samples in the process of crafting poisons, which are more likely to be identified by defenders. Therefore, some clean-label backdoor attacks [42] are also proposed and craft the backdoor samples using the strategy of feature collision presented in Reference [40].

2.4 Advanced Persistent Threats

Advanced persistent threats (APTs) are a series of new defined attack process, in which the attacks are continuous for a long period, such as over years. The workflow of analyzing APTs forms the basis of our analysis framework for adversarial attacks and defenses.

2.4.1 What is APT.

— *Advanced*: APTs are conducted by a group of advanced attackers who are sponsored by some well-established organizations and have access to sophisticated, advanced tools. Conversely, traditional attacks are often performed by a regular attacker.
— *Persistent*: APT attacks generally have long-term goals, and they do not give up on them easily, whereas in traditional attacks, the attackers might choose to quit or change the targets if they encounter seemingly intractable defenses. Further, in APT the attacks are usually sustained for an extended period.
— *Threat*: The threats in APT tend to have little correlation to financial gain, but much to do with competitive advantages and strategic benefits, such as the loss of sensitive data or impediments to critical components.

2.4.2 Lifecycle in APT. To attack the target successfully, the attackers in APT need to go through a complete lifecycle with multiple stages in a sequential manner. There are several different forms of this lifecycle that vary depending on different considerations with respect to generalization or specification. We selected a representative five-stage APT lifecycle model with the goal to undermine critical infrastructure [19]. These stages are as follows.

— *Stage 1: Reconnaissance.* As the first step of attack, the goal of the reconnaissance phase is to learn about the target organization extensively and to gather extensive information that can help the attackers to find the weaknesses in the targets, such as the habits of employees or their favorite websites. The more attackers can understand their targets, the better their success rate.

— *Stage 2: Establish Foothold.* In this phase, the adversaries use the information from the previous stage, Reconnaissance, to exploit vulnerabilities in the target system, such as software bugs or known application vulnerabilities exposed from the well-known vulnerability database. Malware, Spear-phishing, and Watering-hole Attack are usually used in this stage.
— *Stage 3: Lateral Movement.* After the attackers have gained access to the target system, they can try to spread themselves to other systems within the same internal environment via malware or privilege escalation. In this phase, attackers aim to transfer their foothold to other systems to achieve further goals.
— *Stage 4: Impediment.* This phase is characterized by actions to undermine the essential components of the target. That is to say, attackers start to implement actions in this stage, bringing substantial impacts on the target.
— *Stage 5: Post-impediment.* Attackers can continue imposing impediments until the full attack is lifted, which is viewed as one of the actions in Post-impediment. In addition, attackers can delete evidence, such as installed tools and logs for clean exit.

3 ADVERSARIAL ATTACKS

Adversarial attacks present new challenges to deploying deep learning on a large scale. Many research studies have embarked on this journey in recent years. APTs offer a systematical framework for modeling the process of cyber attacks and capturing the real features of different attacks and their inter-relations. We are also interested in understanding the threats of adversarial samples from the cybersecurity perspective. However, the present taxonomies of adversarial attacks are often decided according to individual strategies, while neglecting the relationships between different attacks from a global view. Inspired by APTs, we propose an analysis framework for adversarial attacks that could help to establish a standard in understanding the security problems of deep learning systems. Specifically, we defined a lifecycle for adversarial attacks comprising five stages based on the different attack objectives. The logic of this framework aligns with the APT lifecycle.

3.1 Overview: APT-like Attack Lifecycle

For a systematic understanding of adversarial attacks and to achieve better attacking performance, we need a standard analysis framework through which to explore the vulnerabilities of a deep learning system against adversarial samples. What follows is a five-stage lifecycle of an adversarial attack based on the APT lifecycle. And the methods and objectives in different stages are outlined as follows:

- **Stage 1: Vulnerability analysis.** The first stage contains methods to theoretically analyze the risks of adversarial examples, which shares a similar goal with the Reconnaissance stage of APT—that is, improving the success rate of attacks by learning more knowledge about the target. In an APT, scanning and social engineering methods can be used to help explore the weaknesses in the target system [19].

  Likewise, the robustness can be one of the potential vulnerabilities due to the poor interpretability of DNNs [43]. In the first stage of adversarial attacks, adversaries will conduct a theoretical analysis to investigate the intrinsic sensitivity of DNNs to perturbations [5, 44], understand which factors will influence robustness, and the reasons why a DNN is not robust [45]. Such an analysis can help improve the design of attack methods by exploiting the tensions between standard accuracy and network robustness.
- **Stage 2: Crafting.** Methods utilizing the information from Stage 1 to design general attacks regardless of the target model's structure fall into Stage 2, Crafting. These methods also

share a similar goal with the second stage of APT (Establish Foothold), conducting attacks (like Malware and Spear-phishing) to obtain an entry to the target system based on the information collected in the previous stage [19], which is essential to further impose advanced attacks.

In this stage, some general and fundamental attacks exploiting the unavoidable vulnerabilities of DNNs can be performed based on the exploration in Stage 1. Methods in the crafting stage focusing on the generation of adversarial samples **from scratch** [2, 46] can be treated as the "successful entry" to the target DNNs, while attacks in the next stage, Post-crafting, are designed to further increase the success rate of general attacks or achieve more natural adversarial examples, instead of simply crafting.

- **Stage 3: Post-crafting.** The third stage in adversarial attacks, Post-crafting, simulates the process of the "Lateral Movement" stage in the APT lifecycle where the foothold is expanded to other machines within the target system due to privilege escalation and search for critical components or data [19]. In both APT and adversarial attacks, these methods in this stage can be considered as advanced attacks based on existing "successful entry."

  Post-crafting includes "advanced" attacks working well with only black-box access to DNNs [36, 39] or other attacks considering model-specific features (like the structure of GNN) [47]. They can be thought of as extensions to the general attacks in stage Crafting. Transferability means a black-box attack will generally have a high success rate on unknown DNN [8] and impact more legitimate examples [48]. Model-specific features can empower attackers to design successful attacks in more challenging scenarios [49].
- **Stage 4: Practical Application.** This stage is similar to the Impediment stage of the APT lifecycle [19], as both aim to launch attacks in practical applications, overcome potential problems hindering a successful attack in the real world, and cause actual impacts on the target.

  The adversarial attacks in this stage will deal with some practical applications in both the digital space and the real world, considering the specific features of different domains [35, 50] to further increase an attack's chances of success. The "robustness" of adversarial samples against complex practical environments (such as the noises) will be improved further [51].
- **Stage 5: Revisiting Imperceptibility.** In the final stage of an APT, the adversary erases the evidence of attacks to avoid exposing the traces of attackers and sponsors [19]. Similarly, the attackers in adversarial attacks also desire to stay undetectable at all times.

  In adversarial settings for DNNs, the goal of the adversarial samples is not just to fool the target model, but to ensure the distortions remain imperceptible to humans, which is another underlying requirement for the efficacy of evasion [52]. Otherwise, these perturbed examples might be recognized and discarded by the user of the victim model. Therefore, the final stage of adversarial attacks is Revisiting Imperceptibility, where the objective is to minimize the distortions added while maintaining the attack's success rate [53, 54].

The correspondences between the stages of our framework for adversarial attacks/defenses (see Section 4) and the APT framework are illustrated in Table 1 in an intuitive way. The second column and third column represent the lifecycle with five stages of APT and that of adversarial attacks, respectively. In each row, methods from these two domains will share the similar short-term goals in their lifecycle.

By studying the features and ideas of different adversarial attacks, we can catalog the various adversarial attack methods ranging from theoretical analysis to practical application based on our framework and compose our workflow for increasing the success rate of attacks or broadening the scope of attacks. In the remainder of this section, we review the literature pertinent to each of the

Table 1. Mapping from Five Stages of APT to that of Adversarial Attacks and Defenses

Stage	APT		Adversarial Attacks	Adversarial Defenses
Stage 1	Reconnaissance	→	Vulnerability Analysis	Robustness Certification
Stage 2	Establish Foothold	→	Crafting: Underlying Attack	Anti-Crafting: Training-based Defenses
Stage 3	Lateral Movement	→	Post-crafting: Advanced Attacks	Post-crafting: Purification Defenses
Stage 4	Impediment	→	Practical Application	Application specific Defenses
Stage 5	Post-Impediment	→	Revisit Imperceptibility	Detection Defenses

The second column refers to the lifecycle with five stages of APT. Similarly, the third column and fourth column represent the lifecycle of adversarial attacks and adversarial defenses, respectively. In each row (i.e., one specific stage), methods from different domains share the similar objective. For example, in Stage 1, Reconnaissance aims to scan the vulnerabilities of target system for further attacks, while attackers in adversarial settings try to explore the potential sensitivity of DNNs for perturbations. Additionally, defenses in Stage 1 aim to eliminate the vulnerabilities that can be exploited by adversaries in the reconnaissance.

five stages of the attack lifecycle. Unlike previously published reviews of adversarial attacks, we are the first attempt to identify a lifecycle for adversarial attacks in deep learning. A summary is shown in Figure 3 with the goals of the different stages listed in the last column.

3.2 Stage 1 - Vulnerability Analysis

Most researchers focus on devising attacks that directly and effectively craft adversarial examples. By contrast, few approaches have been developed to quantify the vulnerabilities of adversarial examples, such as a comprehensive measure of robustness, which can be regarded as the power of adversarial examples for one DNN model.

Mahloujifar et al. [45] are interested in the intrinsic nature of robustness and factors that influence it. They investigated why it is challenging to make machine learning models robust. They performed a theoretical analysis for robustness in machine learning classifiers, demonstrating their conclusions that connect robustness and the phenomenon of "concentration of measure" in metric. That is, for any classification model with some initial constant error, if the concentrated metric probability spaces are used, such as Lévy instance spaces, then the model is inherently vulnerable to adversarial perturbations.

Tsipras et al. [5] point to the inherent tension between the standard accuracy and robustness against adversarial perturbations of models. As a complement to Reference [5] and Ding et al. [44] have shown that there are different levels of tradeoffs between clean accuracy and adversarial robustness, which depend on the characteristics of the input data distribution. Based on the conclusions of Tsipras et al. [5] and Ilyas et al. [55] also explored the robust features and non-robust features. They considered non-robust features to be one class of features, which are highly predictive but brittle.

Discussion. The vulnerability of DNNs against adversarial examples might be caused by properties of models or data. For example, the metrics used in the training [45] or the different goals of standard generalization and adversarial robustness [5] would cause constraints for robustness from the perspective of models. In addition, non-robust features [55] might lead to brittle DNNs from the data perspective. Neither of them might be the only cause of this vulnerability. However, it is possible to design effective attacks based on any possible hypothesis. And the details about the hypotheses for existence of adversarial examples are discussed in Section 5.

3.3 Stage 2 - Crafting: General Attacks

In this section, our review begins with a brief description of some seminal works, even though they have been described many times in other surveys, such as References [20, 21]. Subsequently,

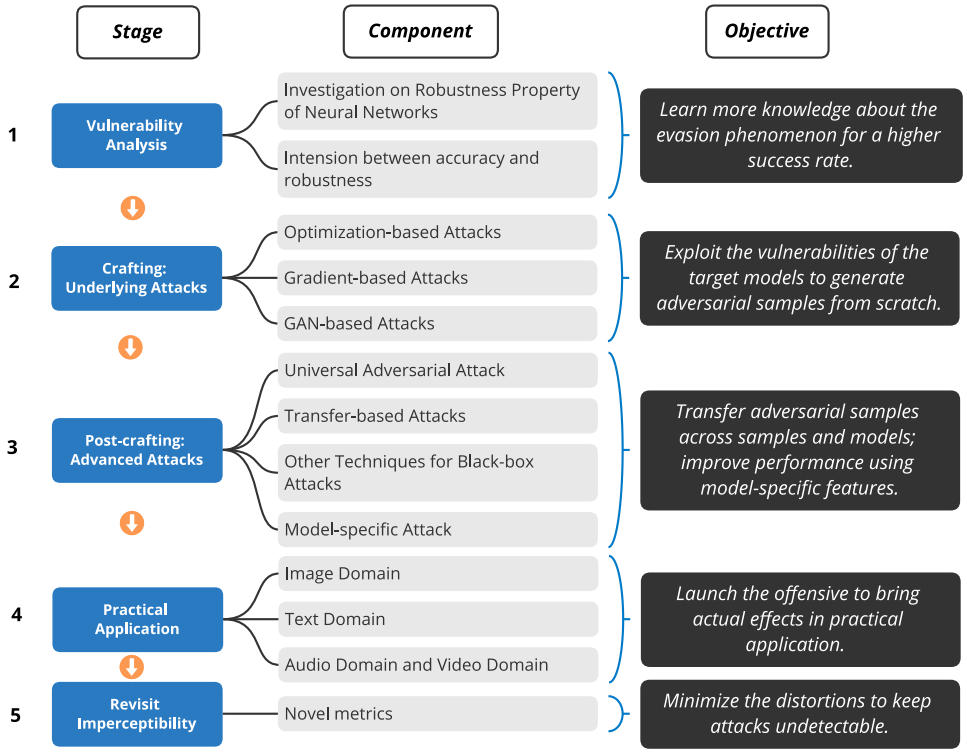

Fig. 3. The five stages of the lifecycle of adversarial attacks are demonstrated here. The component in the second column refers to the types of critical methods used in each stage. And the objective of each stage is summarized in the last column.

we will also illustrate some state-of-the-art general attack approaches, which provide a direction for studying general adversarial attacks. Because numerous relevant works have been published recently, this section is divided into three categories further according to their strategies: Optimization-based Attacks, Gradient-based Attacks, and GAN-based Attacks.

3.3.1 Optimization-based Attacks. As mentioned above, Szegedy et al. [2] first introduced an attack scheme, L-BFGS Attack, against DNNs in 2014. This is widely considered the first study on adversarial attacks in deep learning. Their work formulated how to craft a sample for a targeted label as a searching problem for a minimal distorted adversarial example x'. To further improve the performance of the L-BFGS method, Carlini and Wagner [33] proposed a set of optimization-based attacks, termed the **Carlini and Wagner (C&W)** attacks. Unlike the L-BFGS attack relying on a cross-entropy loss $\mathcal{L}(f(x), t)$, C&W attacks involves a margin loss $\mathcal{L}_m(f(x), t)$ as the loss function, which can be customized by attackers. Several different distance measures $D(\cdot)$ including L_0, L_2, and L_∞ norm can be used by attackers in C&W attacks.

To reduce the size of L_2 distortion and improve the imperceptibility between original samples and the adversarial samples in classification models, Moosavi-Dezfooli et al. [30] proposed an attack algorithm named DeepFool. However, few works have designed algorithms using the L_1 metric to craft adversarial samples, though the L_1 distortion is a distance metric that can account for the total variation. Chen et al. [56] proposed an **Elastic-net attack against DNNs (EAD)**, which was the first to introduce the L_1 norm into an adversarial attack. Their optimization problem is

shown in Equation (3), where $\mathcal{L}(f(x'), t)$ is the target adversarial loss function, and the additional two terms are used to minimize the perturbation in terms of L_1 and L_2 distance in searching:

$$\text{minimize } c \cdot \mathcal{L}(f(x'), t) + \beta \|x' - x\|_1 + \|x' - x\|_2^2 \text{ subject to } x' \in [0, 1]^m. \tag{3}$$

The algorithms mentioned above chose the L_p norm to evaluate the perturbations. By contrast, Zhao et al. [29] used information geometry to understand the vulnerabilities of DNNs to adversarial attacks. They formalized the optimization problem as a constrained quadratic form of the **Fisher Information Metric (FIM)** and presented this novel attack algorithm named **one-step spectral attack (OSSA)** as a way of computing the optimal perturbations with the first eigenvector. Zhang et al. [57] proposed blind-spots attacks, which find some inputs that are far enough from the existing training distribution to fool the target model, because they discovered the adversarially trained network gradually loses its robustness on these data.

3.3.2 Gradient-based Attacks. Although optimization-based L-BFGS attacks achieve high misclassification rates, an expensive linear search method is needed to find optimal hyperparameter c in Equation (2), which has a high computational cost. Thus, Goodfellow et al. [46] proposed a fast one-step method of generating adversarial perturbations called FGSM. This algorithm is described in Equation (4), where $sign(\cdot)$ is the signum function and $\nabla_x(\mathcal{L}(f(x), y))$ represents the gradient of loss w.r.t. x:

$$\Delta x = \epsilon \cdot \text{sign}(\nabla_x(\mathcal{L}(f(x), y))). \tag{4}$$

Because FGSM computes the perturbation with only one backpropagation step of calculating the gradient, it is much quicker at finding adversarial samples than L-BFGS attacks. However, FGSM has a low success rate. To address this shortcoming, Kurakin et al. [4] proposed an iterative version, **Basic Iterative Method (BIM)**. To constrain the adversarial perturbations, BIM adds a clip function (Equation (5)), such that the generated sample is located in the $\epsilon - L_\infty$ ball of the benign image, where x'_i is the intermediate result in ith iteration, and α is the size of the perturbation. In addition to BIM, Dong et al. introduced a momentum optimizer to optimize BIM, which is called **momentum iterative FGSM (MI-FGSM)** [58]. Madry et al. [3] presented the **Projected Gradient Descent (PGD)** attack. PGD replaces the *Clip* function in BIM with the *Proj* function, which is one of the strongest attacks that use the first-order information of target models.

$$x'_{i+1} = \text{Clip}\{x'_i + \alpha \cdot \text{sign}(\nabla_x(\mathcal{L}(f(x'_i), y)))\} \tag{5}$$

Papernot et al. [34] proposed a targeted attack focusing on the perturbations under an L_0 distance metric, called **Jacobian-based Saliency Map Approach (JSMA)**. A Jacobian matrix is used to determine which element is more important for crafting effective adversarial examples. However, generated perturbations by JSMA are greater than that of DeepFool [30]. Based on the idea of DeepFool, Alaifari et al. [37] proposed a novel kind of adversarial attack, ADef, which finds "small" perturbations of the images. Due to the projection nature, iterative algorithms usually lead to large-scale distortions. Chen et al. [59] aimed to address this problem with an attack framework, Frank-Wolfe, which uses momentum mechanisms to avoid projection and leads to better distortions.

In early studies, it was common to generate attack perturbations independently for each specific input based on the loss function. However, Zheng et al. [60] proposed an algorithm called **Distributionally Adversarial Attack (DAA)** [60] to generate deceptive examples by introducing direct dependency between all data points, which is a variant of PGD [3] to maximally increase the generalization risk. Besides, PGD is also shown to lead to overestimation of robustness because of the sub-optimal step-size and problems of the objective loss [61]. Therefore, Croce and Hein [61] proposed a parameter-free version of PGD with an alternative objective function, Auto-PGD, avoiding

the selection of step size for l_2 and l_∞ perturbations. Their extensive experiments showed that Auto-PGD can decrease the accuracy of the target model under multiple existing defenses by more than 10%. In addition, when taking into account l_1-perturbations, PGD is not effective and is weaker than some state-of-the-art l_1 attacks [56]. Croce and Hein [62] analyzed the reason why PGD is sub-optimal under l_1 perturbations and identified the correct projection set, which is computationally feasible. Their method can encourage adversarial training to yield a more robust l_1-model with $\epsilon = 12$ when compared to the original PGD [3].

3.3.3 GAN-based Attacks. Malicious perturbations can lead to some unnatural or not semantically meaningful examples. To craft natural adversarial samples, Zhao et al. [63] presented a framework to craft natural and legible adversarial examples using the GAN in black-box settings. A Generator $\mathcal{G}$ on corpus $\mathcal{X}$ and a corresponding Inverter $\mathcal{I}$ are trained separately by minimizing the reconstruction error of x and the divergence between the sampled z and $\mathcal{I}(\mathcal{G}(z))$. Given an instance x, they searched the perturbations with Inverter in the dense representation of $z' = \mathcal{I}(x)$. And then, they mapped it back to x' with the trained Generator $\mathcal{G}$. The perturbations from the latent low-dimensional z space can encourage these adversarial samples to be valid.

Xiao et al. [64] also proposed a GAN-based attack, AdvGAN, to generate adversarial samples with good perceptual quality efficiently. They added a loss for fooling the target model and another soft hinge loss to limit the magnitude of the perturbation. Once the generator is trained, the perturbations can be generated efficiently for any input, which can potentially accelerate some defensive methods such as Adversarial Training. Based on the architecture of Reference [64], Wei et al. [28] also proposed a GAN-based adversarial attack named **Unified and Efficient Adversary (UEA)** to address problems with high computation costs and the weak transferability of existing methods in image and video object detection. In their method, the generation process only involves the forward networks, so it is computationally fast. In addition, Phan et al. [65] proposed to use GAN to design black-box attacks, improving the transferability of existing attacks.

Discussion. Attack methods discussed in this section show the representative strategies of crafting adversarial examples. Despite the strong performance of optimization-based attacks, most attackers are willing to explore gradient-based attacks, because these kinds of attacks are simpler than their optimization-based counterparts. In addition, efficient attacks can be easily incorporated into defensive techniques against adversarial examples, such as adversarial training, which can increase the efficiency of defenses as an ultimate goal. However, common gradient-based methods need full knowledge of target models, which mainly consist of white-box attacks.

3.4 Stage 3 - Post-crafting: Advanced Attacks

This section introduces some "advanced attack" that can be thought of as extensions to the general attacks mentioned above. These extensions stretch in two directions: horizontally to broaden the scope of the attack and vertically to improve the depth and success rate of attack. Improving the depth and success rate of attacks vertically can be done by combining the common vulnerabilities of DNNs with the model-specific properties.

Improving the scope of the attack expands the influence of adversarial samples to more models or instances. Extending an attack horizontally can be accomplished in three ways: (1) cross-sample transferability of the crafted adversarial perturbations to impact more inputs; (2) cross-model transferability of perturbations to improve their effectiveness on more unknown target models; and (3) some other techniques to attack models in more challenging settings, like a black-box hard-label setting.

3.4.1 Universal Adversarial Attack. Cross-sample transferability captures the ability of a perturbation against one benign sample to be effective against other benign instances within the

same models. General methods aforementioned craft different perturbations for each single sample. Thus, it is not clear about the transferability across the benign samples. Moosavi-Dezfooli et al. [48] first showed the existence of universal adversarial perturbations and provided a systematic algorithm to create these perturbations. At each iteration, they compute the minimal perturbation according to the boundary of the current classification region and then aggregate them into the original input. When a crafted universal perturbation is added to any benign example in the training dataset, all generated adversarial examples are misclassified with a high probability. Sarkar et al. [66] proposed black-box universal adversarial attacks, which can produce targeted misclassification when compared to the initial works [48]. They used a residual generating network to produce an image-agnostic perturbation for each class to misclassify the samples with this perturbation as being from the corresponding class.

Shafahi et al. [67] proposed an efficient optimization-based method to produce the universal perturbations by solving some common problems for DNNs. Specifically, they use stochastic gradient methods to solve the optimization problem of crafting perturbations and introduce a "clipped" version of the cross-entropy loss to mitigate problems caused by unbounded cross-entropy. As a result, their methods dramatically reduce the time needed to craft adversarial examples as compared to Reference [48]. Co et al. [68] proposed to leverage procedural noise functions to generate universal adversarial perturbations. It is simpler to implement, and the smaller search space of procedural noise makes a black-box attack on large-scale applications feasible. Zhang et al. [69] reviewed existing universal adversarial attacks, discussed the challenges, and studied the underlying reasons for why universal adversarial perturbations exist.

3.4.2 Transfer-based Attacks. So far, white-box attacks are based on the assumption that the adversary has access to information such as input data, model structure, gradients, and so on. However, in most scenarios, attackers have little information about models except the input-output pairs. The target models can only be used in a black-box manner. So, black-box attacks are far more common. Transfer-based attacks are probably the most common methods of exploring the cross-model transferability and attacking a black-box target model with the help of white-box substitute models.

Szegedy et al. [2] first described the phenomenon that adversarial examples crafted carefully for one model could be transferred to other models, regardless of its structural properties, like the number of layers. Papernot et al. [39] further explored the property to study how the adversarial samples could be transferred between different machine learning techniques and proposed the first effective algorithm to fool DNN classification models in a black-box manner. They assumed that attackers have no access to the parameters of the classifiers but do have some partial knowledge of the training data (e.g., audios, images) and the expected output (e.g., classification).

To further increase the transferability of adversarial perturbations, ensemble attacks have been a category of crafting transferable perturbations for black-box models. Che et al. [36] proposed Serial-Mini-Batch-Ensemble-Attack, where they consider the process of crafting adversarial samples to be the training of DNNs, and the transferability of the adversarial examples is thought of as the model's generalizability. Phan et al. [65] proposed a GAN-based black-box attack method, called Content-aware Adversarial Attack Generator, which improves on the low transferability of existing attacks in the black-box settings by introducing random dropout.

Domontis et al. [8] provided a comprehensive analysis of transferability for adversarial attacks. They highlighted three metrics: the magnitude of the input gradients, the gradient alignment, and the variability of the loss landscape. In addition, Sharma et al. [70] proposed another factor, perturbation's frequency, from the perspective of perturbations instead of models. They validated adversarial examples are more transferable and can be generated faster when perturbations are constrained to a low-frequency subspace.

To address the weak transferability of black-box attacks (especially under the existing defenses), some state-of-the-art works introduced some novel techniques to improve the cross-model transferability, such as meta learning [71], variance tuning [72], and **feature importance-aware (FIA)** [73].

3.4.3 Query-based Attacks. The performance of the black-box attacks can be influenced by poor transferability in transfer-based attacks using substitute models. In addition to transfer-based attacks, black-box adversaries can use some zeroth-order optimization methods to estimate numerically the gradient through a number of queries, which are denoted as query-based attacks. To avoid using the transferability, a **zeroth order optimization (ZOO)**-based method has been proposed that has high visual quality [74]. However, ZOO relies on the coordinate-wise gradient estimation technique, which demands an excessive number of queries on the target model. As such, it is not query-efficient and rather impractical in the real world. Tu et al. [75] proposed a generic framework for implementing query-efficient black-box attacks, termed as **Autoencoder-based Zeroth Order Optimization Method (AutoZOOM)**. They proposed a scaled random full gradient estimator and dimension reduction techniques (e.g., autoencoder) to reduce the query counts. Ilyas et al. [76] used natural evolutionary strategies to construct an efficient unbiased gradient estimator, which requires far fewer queries than the traditional attacks based on finite-difference.

Square attacks were proposed to further improve the query efficiency and success rate of black-box adversarial attacks, which combines the classical randomized search schemes and heuristic update rule [77]. Yatsura et al. [78] argued that the performance of attacks based on random search depends on the manual tuning of the proposal distributions. Therefore, they formalize square attack as a meta-learning problem to perform automatic optimization, which can circumvent the heuristic tuning and decrease the impact of manual design.

Query-based attacks would be ineffective in real-world scenarios due to the limited information [76]. Brendel et al. [38] proposed a decision-based attack, called a Boundary Attack, which requires less information about or from the models and solely relies on the final decision. For the label-only setting, Ilyas et al. [76] also proposed a concept of discretized score to quantify how adversarial the perturbed image is, which was used to estimate the absent output scores. Likewise, Cheng et al. [79] assumed that the adversary can only observe a final hard-label decision. They reformulated the task as a real-valued optimization problem by binary search. Cheng et al. [80] also optimized their previous work [79] and directly estimated the sign of the gradient rather than the gradient itself, which reduces the number of queries significantly.

The amount of queries for query-based attacks has decreased from millions to less than a thousand [81]. Maho et al. [81] proposed a geometrical approach, SurFree, based on the decision in a black-box setting. They bypassed the usage of surrogate of the target model and estimation of the gradient. Ilyas et al. [82] introduced a framework unifying a previous black-box attack methodology. They proposed bringing gradient priors into the problem to further improve the performance of untargeted attacks. Narodytska and Kasiviswanathan [83] proposed a black-box attack in an extremely limited scenario where only a few random pixels can be modified when crafting adversarial examples. They found that a tiny number of perturbed pixels is sufficient to fool neural networks in many cases. Su et al. [84] proposed the one-pixel attack and restricted the perturbation to only one pixel. They used differential evolution to find the optimal position of the perturbation and modified its RGB value to fool the target model.

3.4.4 Model-specific Attacks. Designing an attack that exploits transferability to invalidate more models can be viewed as a horizontal extension to underlying attacks. Beyond this, there are studies on vertical extensionsthat exploit the properties of specific models (like the structure of GNN) to increase the chance of an attack's success. For example, first-order optimization can not

be applied directly to attacks on node classification tasks using edge manipulations for GNNs because of the discrete structure of graphs. Xu et al. [47] presented an approach to generate topology attacks (i.e., edge manipulation attacks) via convex relaxation, which empowers the gradient-based attacks that can be applied to GNNs. Chang et al. [49] provided a general framework, Graph Filter Attack, to attack graph embedding models in restricted black-box setting without requiring information in Reference [47].

The indifferentiable operations in a **Deep product quantization network (DPQN)** lead to one of the challenges to attack DPQN-based retrieval systems. To avoid the backpropagation, Feng et al. [85] proposed to formulate the generation problem as a minimization of similarity between the original query and the adversarial query. Tsai et al. [86] proposed an attack against a special network for point cloud classification. In addition, PGD might not perform as well on the **Binarized Neural Networks (BNNs)** because of their discrete and non-differentiable nature. Therefore, Khalil et al. [87] formulated the generation of adversarial samples on BNNs as a mixed integer linear programming problem and proposed integer propagation to tackle the intractability.

Moreover, Chhabra et al. [88] first investigated the adversarial robustness of unsupervised learning algorithms like clustering. Through their strategy, perturbing only one sample can lead to the perturbation of decision boundaries between clusters. Reinforcement learning often adopts some self-organization techniques to develop self-managed complex distributed systems [89, 90]. Huang et al. [91] showed the impact of existing adversarial attacks on trained policies in reinforcement learning. They applied FGSM to compute adversarial perturbations for policies. Wu et al. [92] focus on adversarial attacks in reinforcement learning by training an adversarial agent to effectively exploit the vulnerability of the victim without manipulating the environment.

Discussion. There are two challenges in designing adversarial examples for DNNs, limited knowledge and special properties of model. Universal adversarial perturbations usually generalize well across different classification models [48, 68], which can also be used to address limited knowledge. Though transfer-based attacks do not rely on the detailed information of models, the adversary needs to have some partial knowledge of the training data. Transfer-based attacks are prone to suffer from low success rates due to the lack of adjustment procedures for information from surrogate models. Query-based attacks usually achieve higher success rates while they are likely to lead to an enormous number of queries [38, 75, 79, 80]. P-RGF [93] combines transfer-based methods and query-based methods. The transfer-based prior from the surrogate model is utilized to query the target model efficiently, which simultaneously guarantees the attack success rates and query efficiency.

3.5 Stage 4 - Practical Applications

This section reviews adversarial attacks in multiple domain, including image domain [32, 35, 50, 94, 95], text domain [96–98], audio domain [1, 31, 51, 99–101], and systems with streaming input [92, 102]. For applications in different domains, there are various challenges in the implementation of adversarial attacks. For instance, contextual information (in object detection systems) and rigid lexical and syntactical constraints (in code authorship attribution) will prevent from successful generation of adversarial examples.

3.5.1 Image Domain and Video Domain. Eykholt et al. [50] extended an existing algorithm for image classification to object detection domain. They modified the adversarial loss functions to minimize the probability of the target object appearing in the scene. Zhang et al. [32] conducted an experimental study to attacking object detectors for vehicles in the physical world. They tried to learn a camouflage pattern and painted the pattern on the vehicles, finding it could hide the vehicles effectively. Considering the limitations of static adversarial patches, Lovisotto et al. [94] proposed a novel method of generating physically robust real-world adversarial examples through

a projector, which can enhance the robustness of patch-based adversarial examples by increasing the non-printability score.

Sensors are fundamental to the perception system of Autonomous Vehicles. Unlike camera-based perception, only a few papers touch on the feasibility of adversarial attacks on the sensor inputs of LiDAR perception. Cao et al. [35] conducted the first study on the security of LiDAR-based perception against adversarial perturbations by formulating adversarial attacks as an optimization problem. Hamdi et al. [95] also demonstrated that semantic attacks, including changes in camera viewpoint and lighting conditions, are more likely to occur naturally in autonomous navigation applications. Thereby, they proposed a semantic attack based on GAN, treating the process of mapping the parameters into environments as a black-box function.

Zhao et al. [1] provided systematic solutions to craft robust adversarial perturbations for practical object detectors at longer distances and wider angles. Wei et al. [101] focused on the adversarial samples in the video domain, which differs from the images domain given the temporal nature of videos. They leveraged the temporal information to improve the attacking efficiency and proposed the concept of propagating perturbations. A heuristic algorithm to further improve the efficiency of the method is in Reference [79].

3.5.2 Text Domain. Liang et al. [96] applied adversarial attacks to DNN-based text classification. Like FGSM, they also leveraged a cost gradient to generate the adversarial examples, while keeping the text readable. They identified important text items like hot training phrases according to the gradients and proposed three strategies, including insertion, modification, and removal, to manipulate these important items. Finlayson et al. [18] reviewed the adversarial behaviors in medical billing industry, illustrating the influences on fraud detectors for medical claims.

The peculiarities like code layout in the code usually can be used in the tasks to identify authorship information, also called authorship attribution. Quiring et al. [97] proposed the first black-box adversarial attack to forge the coding style by combining compiler engineering and adversarial learning. Other than authorship attribution, it is more difficult to generate robust adversarial samples in source code processing tasks due to the constraints and discrete nature of the source domain. Zhang et al. [98] treated adversarial attacks against code processing as a sampling problem and proposed the Metropolis-Hastings Modifier algorithm, which can craft a sequence of adversarial samples of source code with a high success rate.

3.5.3 Audio Domain. Yakura and Sakuma [51] proposed a special over-the-air condition to describe the difficulties of attacking practical **Automatic Speech Recognition (ASR)** systems, where the audio adversarial sample is played by the speaker and recorded by a device. In this scenario, such attacks can be impacted by reverberation and noise from the environment. Hence, they simulated the transformations caused by replaying the audio and incorporated them into adversarial audio samples. However, several hours are needed to craft just one adversarial sample. Liu et al. [99] proposed weighted-sampling adversarial audio examples, which can be computed at the minute level.

Zhang et al. [100] focused on the non-negligible noise introduced by previous works attacking ASR like Reference [51]. This noise can influence the quality of the original audio and reduce the robustness against a defensive detector by breaking temporal dependency properties. They proposed to extract Mel Frequency Cepstral Coefficient features of audio instances. For some ASR tasks with combinatorial non-decomposable loss functions, gradient-based adversarial attacks are not amenable for them [103]. Usually, a differentiable surrogate loss function is required in this case, while the poor consistency might affect the effectiveness of adversarial examples significantly. Houdini [103] is proposed to be tailored for these task losses to generate effective adversarial examples with high transferability, which can be used in the black-box scenario.

Discussion. In the practical tasks based on DNNs, it would be harder to implement adversarial attacks, because various domain-specific peculiarities and challenges have to be considered. Compared to conventional image classifiers, contextual information and location information in object detection can be used to prevent mispredictions [50]. Some factors in the diverse environments also affect the success rate of attacks, including noise and reverberation from playback in ASR [51, 100], changes in camera viewpoint and lighting conditions [95]. Likewise, generating samples of source code would also encounter rigid lexical and syntactical constraints [97].

3.6 Stage 5 - Revisit Imperceptibility

Adversaries are required to consider the magnitude of their adversarial perturbations and their imperceptibility. The commonly used L_p distortion metrics, including L_0, L_2, and L_∞ norm, are objective and can be detected easily by human eyes. Thus, it may not be optimal to adopt these metrics to evaluate the similarity between benign and adversarial samples. As mentioned by Xu et al. [52], attacks aiming to generate perturbations with a small L_0 norm can cause the L_∞ norm to be very large. Therefore, some researchers have started to develop non-L_p metrics to gauge the distortions over samples to explore the exact imperceptibility for humans.

As opposed to the studies that directly manipulate the pixel values to generate adversarial examples, Xiao et al. [53] proposed the **spatially transformed adversarial example optimization method (stAdv)**. StAdv avoids modifying the pixel values and instead changes the positions of the pixels. Specifically, the pixel in the adversarial image can be synthesized using the 4-pixel neighbors of its corresponding pixel in the original image. As a result, it can craft perceptually realistic examples and preserve the semantics of real instances. Xu et al. [52] also pointed out that the L_p metric is neither necessary nor sufficient, because no single measure can be perfect for human perceptual similarity. They proposed a structured attack, which explores the concept of group sparsity. In their approach, an input image is divided into sub-groups of pixels, and the corresponding group-wise sparsity would be penalized. Like Xu et al. [52] and Liu et al. [54] focused on the image formation process, developing a novel physically based differentiable renderer to perform perturbations in the underlying image formation parameter space. The process changes the pixels to an alternative color.

Discussion. Most studies in the image domain use L_p norms to measure the distortions, and information about spatial structures of images is more likely to be ignored. Despite the simplicity of pixel norm-balls for perturbations, they have been shown not to align with human perceptual similarity well [52]. Some structural metrics, such as positions of the pixels [53] and group sparsity [52], can be introduced, which can help search most effective adversarial examples under the constraints of perturbation magnitude.

The attack methods in different stages of the proposed framework are presented in Table 2. However, the methods falling into Stage 1 (Vulnerability Analysis) are not contained in this table, because most of them only focus on exploring the reason for vulnerability, and no attack schemes are provided in these papers. The attacks are presented along with the information about the threat model, including attacker's goal and knowledge. Specifically, untargeted attacks are strictly less powerful than targeted attacks and can be regarded as one simple form of running a targeted attack for one random target. In turn, some untargeted attacks can be adapted to the targeted version easily. Therefore, the untargeted attacks in this table only mean that authors focused on the untargeted settings in writing. Moreover, many black-box attacks can be achieved by running white-box attacks on substitute models. The white-box attacks in Table 2 also mean that authors mainly focused on a white-box attacker in their work, instead of its inability to impede a model in a black-box scenario.

Table 2. Catalog of Adversarial Attacks Following the Analysis Framework Proposed and the Performance Demonstrated

Stage	Attacks	Attack Strategy	Attacker Goal		Attacker Knowledge		Dataset
			Targeted	Untargeted	Black-box	White-box	
2	L-BFGS [2]	Optimization-based	✓	✓	-	✓	M, IN
2	C&W [33]	Optimization-based	✓	-	-	✓	M, C-10, IN
2	OSSA[29]	Optimization-based	-	✓	✓	✓	M, C-10, IN
2	Blind-spots [57]	Optimization-based	-	✓	-	✓	M, FM, C-10
2	DeepFool [30]	Optimization-based	-	✓	-	✓	M, C-10, IL12
2	FGSM [46]	Gradient-based	✓	✓	-	✓	M, C-10
2	BIM [4]	Gradient-based	✓	✓	✓	✓	M, C-10
2	MI-FGSM [58]	Gradient-based	✓	✓	✓	✓	IL12
2	PGD [3]	Gradient-based	✓	✓	✓	✓	M, C-10
2	Auto PGD [61, 62]	Gradient-based	✓	✓	-	✓	M, C-10, C-100, IN
2	JSMA [34]	Gradient-based	✓	-	-	✓	M
2	Frank-Wolfe [59]	Gradient-based	✓	-	✓	✓	M, IN
2	ADef [37]	Gradient-based	✓	✓	-	✓	M, IN
2	DAA [60]	Gradient-based	-	✓	-	✓	M, FM, C-10, IN
2	LatentGAN [63]	Model-based	-	✓	✓	-	M, LSUN, TE
2	AdvGAN [64]	Model-based	✓	✓	✓	✓	M, C-10, IN
2	UEA [28]	Model-based	✓	-	✓	-	PASCAL, IN
3	Universal Examples [48]	Cross-sample	-	✓	-	✓	IN, IL12
3	UniAdvTraining [67]	Cross-sample	-	✓	-	✓	C-10, IN
3	Procedural Noise [68]	Cross-sample	-	✓	-	✓	IN, IL12, COCO
3	UPSET [66]	Cross-sample	✓	-	✓	-	M, IN
3	Substitute [39]	Transfer-based	-	✓	✓	-	M, GTSRB
3	SMBEA [36]	Transfer-based	✓	-	✓	-	CS, F, GGS, LSUN
3	CAG [65]	Transfer-based	✓	-	✓	✓	C-10, IN
3	Explain Transferability [8]	Transfer-based	-	✓	✓	✓	M, Drebin
3	Using Frequency [70]	Transfer-based	✓	✓	✓	✓	IN
3	Meta Gradient [71]	Transfer-based	✓	✓	✓	-	C-10, IN
3	FIA [73]	Transfer-based	✓	✓	✓	✓	IN
3	Variance Tuning [72]	Transfer-based	✓	✓	✓	✓	IN
3	Boundary Attacks [38]	Query-based	✓	✓	✓	-	M, C-10, IN
3	SurFree [81]	Query-based	-	✓	✓	-	M, IN
3	AutoZoom [75]	Query-based	✓	-	✓	-	M, C-10, IN
3	Bandit Optimization [82]	Query-based	-	✓	✓	-	C-10, IN
3	Query-efficient [76, 79, 80]	Query-based	✓	✓	✓	-	M, C-10, IN
3	P-RGF [93]	Query-based	-	✓	✓	-	IN
3	Square Attack [77, 78]	Query-based	✓	✓	✓	-	M, C-10, C-100, IN
3	A Few Pixels [83, 84]	Query-based	✓	-	✓	-	M, C-10, SVHN, IN, STL10
3	Topology Attacks [47]	Model-specific	✓	✓	-	✓	Cora, Citeseer
3	GF-Attacks [49]	Model-specific	-	✓	-	✓	Cora, Citeseer, Pubmed
3	PQ-AG [85]	Model-specific	-	✓	✓	✓	C-10, NUS-WIDE
3	PointNet++ [86]	Model-specific	✓	✓	-	✓	ModelNet40
3	IProp [87]	Model-specific	✓	✓	-	✓	M, FM
3	Clustering [88]	Model-specific	✓	-	✓	-	U-HD, M, U-WS, MHP
4	Image Domain [32, 50]	Application	-	✓	✓	✓	Unreal
4	Image Domain [35]	Application	✓	-	-	✓	Velodyne
4	Text Domain [96]	Application	✓	-	✓	-	-
4	Code Domain [97]	Application	✓	✓	✓	-	Google Code Jam
4	Code Domain [98]	Application	-	✓	✓	-	Open Judge
4	Audio Domain [51, 99, 100]	Application	✓	-	-	✓	CSN1, TTS / MCV/LS
4	Houdini [103]	Application	✓	✓	✓	✓	CS, LS
4	Video Domain [1, 101]	Application	✓	✓	-	✓	COCO / UCF101
4	Video Domain [31]	Application	-	✓	✓	-	UCF101, HMDB-51
4	streaming input [102]	Application	-	✓	✓	-	Speech commands[104]
5	stAdv [53]	Imperceptibility	✓	-	-	✓	M,C-10, IN
5	StrAttack [52]	Imperceptibility	✓	-	-	✓	M,C-10, IN
5	Parametric Norm [54]	Imperceptibility	✓	✓	✓	✓	C-100

In the table, the following abbreviations have been used: M for MNIST, FM for FashionMNIST, C for CIFAR, IN for ImageNet, IL12 for ILSVRC2012, TE for Textual Entailment, COCO for Microsoft Common objects in context, CS for Cityspaces, F for Facades, GGS for Google Satellites, U-HD for UCI Handwritten Digits, U-WS for UCI Wheat Seeds, MHP for MoCap Hand Postures, CSN1 for Cello Suite No.1, TTS for To The Sky, MCV for Mozilla Common Voice dataset, and LS for LibriSpeech.

4 ADVERSARIAL DEFENSES

Countermeasures against the adversarial attacks mentioned above eliminate some of the risks, and the more vulnerabilities that can be stemmed, the greater the likelihood that deep learning techniques can be deployed on a large scale. To this end, we have developed a defensive lifecycle akin to the framework above for attacks.

4.1 Overview: APT-like Defense Lifecycle

Our analyzing framework for adversarial defenses provides a standard procedure through which to improve a model's robustness against adversarial samples. Like the attack lifecycle, the defense lifecycle also has five stages.

- **Stage 1: Robustness Certification.** To a certain extent, this stage of adversarial defenses can be considered as one where the vulnerability of DNNs against adversarial examples is thoroughly eliminated. As a result, the "scanning" for vulnerabilities of DNNs will fail under the certified robustness guarantees. This stage consists of provable defenses with theoretical guarantees [105, 106]. These can theoretically certify a DNN model's robustness, i.e., the ability of keeping prediction unchanged for small modifications of the input. However, as mentioned, this expectation may be unrealistic for some complex models.
- **Stage 2: Anti-crafting.** This stage addresses the unavoidable complexity and difficulties associated with providing a provable guarantee of robustness for DNNs. Hence, defenders must change their objectives to that of preventing adversarial samples from being generated using attack strategies in Crafting stage of attack lifecycle. The focus here is on the training phase of the target model and trying to develop robust models for which it is difficult to find effective adversarial examples [3, 107]. Training robust models by injecting more augmented data or distillation techniques can ensure that general attacks are ineffective.
- **Stage 3: Post-crafting.** In the third stage of adversarial defenses, defenders assume that the adversarial samples have been crafted successfully. As a result, they have to focus on the scenarios where adversarial samples are inevitable in the training phase and make efforts to prevent the further damage caused by existing vulnerabilities. Defenses covered in this stage can be thought of as reactive defenses. They can modify the architecture of DNNs [108, 109] or deploy preprocessing methods to transform them into legitimate samples to mitigate their impact [110, 111].
- **Stage 4: Application-specific Defenses.** Adversarial attacks in Stage 4 of the attack lifecycle would exploit the features of specific scenarios or special data structures. Thereby, they can be mitigated and defended from the level of applications with some unique properties. Some domain-specific features and factors in the physical world are known to be resistant to adversarial examples [112, 113]. These techniques can be thought of as countermeasures to attacks launched on specific practical applications. They have poor generalization to be transferred to other scenarios but they can achieve high robustness.
- **Stage 5: Detection Defenses.** In adversarial settings, attackers in the Revisiting Imperceptibility stage will be trying to remove evidence of their foul play, avoiding being recognized by humans. In that case, defenders should attempt to identify the malicious adversarial examples by designing effective detectors to circumvent those efforts of attackers [114]. These detecting techniques constitute the final stage of defense lifecycle, Detection Defense. If the defenses in the previous stages have failed to prevent the adversarial samples, then the best strategy left is to minimize their impact on the final predictions.

Table 1 also describes the correspondences between these stages for defenses and the APT framework. It is worth noting that despite the similarity in the stages of the attack and defense lifecycle,

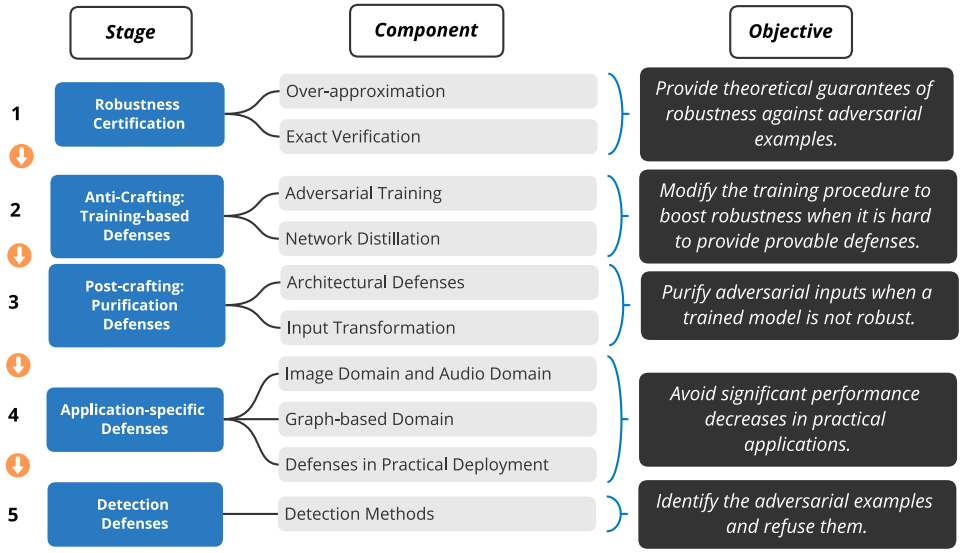

Fig. 4. The five stages of the lifecycle of adversarial defenses are demonstrated here. The component in the second column refers to the types of critical methods used in each stage. The objective of each stage is summarized in the last column.

they do not strictly have a one-to-one correspondence. The similarity only results from the similar short-term objective in the corresponding stages in attacks and defense lifecycle. Moreover, the proposed attack/defense lifecycle presenting a sequence of different attack/defense strategies does not mean to view each strategy in isolation. On the contrary, the lifecycle helps us consider different strategies as a whole, where sometimes a "one versus some" could be used to resist attacks in different stages, and other times a "some versus one" defensive strategy from multiple stages could be integrated to resist one attack. The defenses against adversarial attacks are classified into different stages according to their defensive goals. The components and objectives of the defensive methods in different stages are summarized in Figure 4.

4.2 Stage 1 - Robustness Certification

Most defenses are only validated experimentally; they are not proven to be effective with a theoretical guarantee of error rates. Despite their excellent techniques, their efficacy was short-lived, as the systems discussed were simply attacked again with more aggressive adversarial attacks. Provable defense guarantees falling into Stage 1 are the most effective defensive methods, and they will always work, regardless of types of attacks.

Weng et al. [105] proposed the first attack-independent robustness metric to estimate the lower bound of the minimum adversarial perturbations. They converted the evaluation of robustness into a local Lipschitz constant estimation problem and provided a theoretical justification. Their metric, CLEVER, has been corroborated to be computationally feasible. However, Goodfellow [115] has subsequently reported that the lower bound estimated by CLEVER is incorrect. Ruan et al. [106] proposed a global robustness problem as a generalization of the local robustness problem. Global robustness is defined as the maximal safe radius with the lower bounds and upper bounds over a test dataset, while the local robustness represents the safe radius for one input. Their method was the first algorithm to provide a bound of robustness for a Hamming distance (L_0). Yu et al. [116] proposed a quantitative evaluation metric of robustness regardless of the datasets and attack methods.

Raghunathan et al. [117] proposed a certifiable defense for two-layer neural networks in adversarial settings. However, the convex relaxations in Reference [117] could not scale to large networks. To adapt similar certification methods to deeper networks, Wong and Kolter [118] proposed to construct a convex outer approximation for the activation values as an adversarial polytope against the norm-bounded adversarial perturbations. Sinha et al. [119] defend against adversarial perturbations from the perspective of a distributionally robust optimization. Xiao et al. [120] also focused on the intractability of exact verification problems for adversarial robustness. They proposed the idea of co-design to train neural networks, which aligns the model training with verification and ensures that robust models are easy to verify. Tjeng et al. [121] proposed a **Mixed-Integer Linear Programming (MILP)** verifier to address the verification for piecewise-linear networks, which was considered as a mixed-integer program. Singh et al. [122] combined over-approximation techniques with MILP solvers and proposed a system called RefineZono that chooses neurons to refine their bounds. Their system improves the precise loss for large networks and has faster verification than the work of Tjeng et al. [121].

Discussion. The defenses demonstrated in this section provide certifications of the robustness of machine learning models. In other words, they provide a theoretical guarantee against adversarial samples, which can be considered the strongest defenses. However, incomplete robustness verifiers based on over-approximation methods, like Reference [117], can suffer from a loss of precision when scaled to DNNs. And complete verifiers that leverage MILP usually lack scalability. Therefore, these techniques are too complex and have poor applicability for DNNs.

4.3 Stage 2 - Anti-crafting: Training-based Defenses

To avoid general attacks, we need defenses that can really improve the robustness of the models, rather than some preprocessing techniques or detection strategies for abnormal samples. So, the defenses in Training-based Defenses can be thought of as proactive defenses consisting of Adversarial Training and Network Distillation.

4.3.1 Adversarial Training Techniques. Adversarial Training is a simple and common method of decreasing the test error for adversarial examples by incorporating crafted examples into the training data. Goodfellow et al. [46] proposed an adversarial training method, where adversarial samples are generated using FGSM and then injected them into the training dataset. Adversarial training can promote regularization for DNNs. However, although these adversarially trained models have robustness against one-step attacks, they are still easy to be fooled by iterative attacks. Madry et al. [3] subsequently proposed adversarial training with adversarial examples crafted by PGD attacks. They focused on the "most adversarial" sample in the L_∞ ball around the benign sample. Thus, with this method, universally robust models can be developed against a majority of first-order attacks.

However, adversarial training still has some limitations. For example, because the process of generating each adversarial sample involves an iterative attack [3], adversarial training usually carries a high computational cost, which limits its practicality for large datasets. In addition, the effectiveness of adversarially trained models has been shown to be influenced by some factors, such as other L_p adversaries [107], more complex datasets like CIFAR [123, 124], and the perturbations occurring in the latent layer [125]. Another problem in adversarial training is a decrease in generalization [126, 127].

4.3.2 Distance Metrics and Latent Robustness. Li et al. [107] proposed an improvement for adversarial training. They introduced Triplet Loss (one popular method in Distance Metric Learning) to the adversarial training, which enlarges the distance in embedding space between the

adversarial examples and other examples. By incorporating a regularization term, this new algorithm effectively smooths the boundary and improves the robustness.

Most previous works focus on the robustness of the input layer of DNNs. Kumari et al. [125] found that the latent layers of the robust adversarial-trained model were still vulnerable to perturbations, though the input layer had high robustness. Based on this observation, they proposed Latent Adversarial Training, a fine-tuning technique, to improve the efficacy of adversarial training.

4.3.3 Complex Tasks. Moreover, Buckman et al. [123] aimed to adapt adversarial training techniques to complex datasets where PGD-based Adversarial Training [3] is usually ineffective. Based on the hypothesis in Reference [46] that the over-generalization in models that are too linear leads to the vulnerabilities toward adversarial samples, they proposed to leverage the quantization of inputs to introduce a strong non-linearity. What they found was that combining them with the adversarial training increases adversarial accuracy. However, applying quantization alone can be broken easily. Thereby, Cai et al. [128] proposed curriculum adversarial training technique to improve the resilience of adversarial training and increase the performance on complex tasks. Specifically, they used a weak attack to train the model first and then increased the strength of the attack gradually until it reached an upper bound. Liu et al. [124] also addressed scaled up problems with complex datasets by combining the adversarial training with Bayesian learning.

4.3.4 Generalization. Considering the decreasing performance of adversarial training in the face of random perturbations from other models, Tramèr et al. [126] proposed **Ensemble Adversarial Training (EAT)** to improve the generalization of defenses in the face of different attacks. The premise is to generate and transfer some one-step adversarial examples from other pre-trained models to augment the training data. Unlike the adversarial training in References [3, 46], EAT decouples the training phase from the generation process of adversarial examples. Similarly, Na et al. [127] also use already-trained models to design adversarial training, referred to as cascade adversarial training. Farnia et al. [129] proposed using regularization techniques to increase the adversarial test performance for adversarial training. They provided a theoretical analysis for the improving generalization with DNNs after introducing a computationally efficient regularization technique (spectral normalization) that significantly decreases generalization errors. Song et al. [130] also aimed to increase the generalization of existing methods based on adversarial training to resist adversarial perturbations from the perspective of domain adaptation.

4.3.5 Network Distillation. Distillation is a popular transfer learning method, where smaller target models can be trained based on larger source DNNs. The knowledge of the source models can be transferred to the target models in the form of confidence scores. Papernot et al. [131] proposed the first defense method using network distillation against adversarial attacks in DNNs. Here, the computer could not find the gradient of the target model and, therefore, the gradient-based attacks would not work. Goldblum et al. [132] studied the distillation methods for generating robust target neural networks. They observed that the adversarial robustness could be transferred from a source model to a target model, even if the source model had been trained using clear images. They also proposed a new method, called **Adversarially Robust Distillation (ARD)** for distilling robustness onto smaller target networks. ARD encourages target models to imitate the output of their source model within an $\epsilon-$ball of training samples, which is essentially an analog of adversarial training.

Discussion. The strength of adversarial training with the worst-case perturbations is satisfactory. Adversarial training with PGD is a state-of-the-art defense, and this technique is easy to apply when compared with the certified robustness [121]. However, adversarial training requires re-training models and much computational resources, which causes the poor scalability to DNNs.

4.4 Stage 3 - Post-crafting: Purification Defenses

Training-based defenses might be impractical as a category of recourse-consuming defenses, especially with a large amount of training data. In this stage, defenders aim to mitigate the risks of adversarial samples caused by the existing vulnerabilities of DNNs by preprocessing the input.

4.4.1 Architectural Defenses. Due to the limitations of adversarial training, some defenses follow a different strategy of modifying the architecture of models without the need for additional training. Controlling Lipschitz constants layer-wise [108] and decreasing the invariance of DNNs [133, 134] can improve their robustness against adversarial perturbations. However, methods based on controlling Lipschitz constants are usually weaker than adversarial training [3]. Along these lines, Qian et al. [108] proposed a new method of controlling the Lipschitz constants of networks by modifying the regularization and loss functions. Schott et al. [109] highlighted the limitations of adversarial training especially for non-L_∞ perturbations and focused on the influence of unrecognizable images, modeling the class-conditional distributions by means of a Bayesian model.

Neklyudov et al. [133] also demonstrated that the variance layers they proposed (a different family of the stochastic layer) could provide higher robustness against targeted adversarial attacks. Jacobsen et al. [134] provided a novel view that the failures of machine learning models result from an excessive invariance to changes that are semantically meaningful. To address this issue, they modified the loss function by means of invertible networks, which can provably reduce the unwanted invariance.

4.4.2 Input Transformation. The strategy of these defenses is to directly denoise the input instances and transform them into legitimate samples before they are fed into the target model. Song et al. [110] empirically evaluated a hypothesis that the adversarial examples usually lie in low probability even though the perturbations are very small. Thereby, they chose a neural density model to model image distributions to detect adversarial examples effectively, which can compute the probabilities of all images and the probability density of one input. Further, to mitigate the impact of adversarial perturbations, they proposed PixelDefend to purify adversarial examples by searching a probable image within a small deviation to the original input with a true label. Samangouei et al. [111] proposed a GAN-based denoising technique for adversarial examples regardless of the types of model. They leveraged the expressive capability of the generator to model the distribution of clean images. In addition, randomization is a commonly used technique in adversarial defenses [135, 136].

Discussion. As reactive defenses, purification including architectural defenses and input transformation are compatible with other defenses to improve the robustness further. Specifically, variance layers [133] can be combined with the idea of ensembles [126]. Input transformation techniques are model-agnostic and are regardless of whether an adversarial attacks appears or not [110, 111, 135]. Therefore, most of them can be combined with the other proactive defenses.

4.5 Stage 4-Application-specific Defenses

Some domain-specific features and factors in the physical world are known to be resistant to adversarial examples in various applications, which can be used to mitigate the effectiveness of the malicious examples.

4.5.1 Image Domain and Audio Domain. Face recognition is one of the simplest applications of image classifiers. Goswami et al. [113] analyzed the impacts of adversarial perturbations from preprocessing on face recognition tasks and demonstrated that simple distortions such as black grid lines could decrease the face verification accuracy. They proposed a countermeasure to detect the adversarially modified faces by evaluating the response from hidden layers of target models.

For the identified faces with distortions, they exploited selective dropout to preprocess them before the recognition, which can rectify them to avoid the significant performance decrease.

Guo et al. [112] pointed that input transformations using JPEG compression have not been verified to be effective for strong adversarial attacks like FGSM. Thereby, they aimed to increase the effectiveness of input transformation-based defenses while preserving the necessary information for classification. They provided five transformations that can surprisingly defend existing attacks when the training data for DNNs was processed in a similar way before training. Xiang et al. [137] proposed a general defense framework against localized adversarial patches in the physical world, PatchGuard, which is compatible with any CNN with small receptive fields. Yang et al. [138] explored the potentials of audio data towards mitigating adversarial inputs and demonstrated the discriminative power of temporal dependency in audio data against adversarial examples. Hussain et al. [139] also studied the effectiveness of audio transformation-based defenses for detecting adversarial examples.

4.5.2 Graph-based Domain. In addition, the structure of training data in the graph-based domain can be used in the design of adversarial defenses. Svoboda et al. [140] proposed new paradigms to develop more advanced models directly with higher robustness. The family of deep learning models on graphs named Peer-regularized Networks can exploit the information from graphs of peer samples and perform non-local forward propagation. Wu et al. [141] investigated the defense on graph data and pointed out that the robustness issue of GCN models was caused by the information aggregation of neighbors. They proposed a defense that leverages the Jaccard similarity score of nodes to detect adversarial attacks, because these attacks can improve the number of neighbors with poor similarity. Yang et al. [27] paid attention to the rumor detection problems on social media using camouflage strategies. They proposed a graph adversarial learning method to train a robust detector to resist adversarial perturbations, which increased the robustness and generalization simultaneously. Goodge et al. [142] focused on unsupervised anomaly-detecting autoencoders and analyzed its adversarial vulnerability.

4.5.3 Defenses in Privacy. Apart from the adversarial perturbations crafted maliciously to mislead models, there are some "benign" perturbations that can cause positive influences to existing tasks based on DNNs by mitigating privacy concerns in deployment. Privacy attacks in machine learning can often be addressed using **Differential Privacy (DP)** technique, which has been well studied [143–145]. For example, due to the advantageous properties of differential privacy, it can also contribute to stabilize learning [146] or build heuristic models for game-theoretic solutions [147, 148]. Interestingly, benign adversarial perturbations can be used to build defenses to protect privacy in machine learning, such as membership privacy of training data [149, 150].

Discussion. Most works have been focusing on the adversarial examples in the image domain. In physical world, when conventional extensions of existing attacks for images fail to remove adversarial threats effectively in other domain, defenders can exploit some domain-specific features to resist adversarial examples. For example, due to the subtle effects of input transformation defenses from the image domain in speech recognition systems, temporal dependency in audio data can be used to improve the effectiveness of detection [138]. However, domain-specific features are usually used to improve the strength of input transformation defenses [112, 137, 138, 141] to preprocess examples, while few of them, such as information of peer samples in graphs [140], can help train model with higher robustness.

4.6 Stage 5 - Detection Defenses

Some metrics of DNNs might be correlated with the adversarial examples, such as the dimensional properties [114], feature attribution scores [151], and distances between adjacent classes [152].

These metrics can be used to detect whether adversarial samples exist or not. Carlini and Wanger [114] showed us the limitations of previous detection-based defenses. Subsequently, Ma et al. [153] considered measures of intrinsic dimensionality to effectively detect the adversarial samples. They proposed a metric to characterize the dimensional properties of the adversarial regions in which adversarial examples lie, referred to as local **intrinsic dimensionality (LID)**. They revealed that adversarial samples had higher LID characteristics than normal samples. He et al. [152] proposed an attack method, OPTMARGIN, which can evade the defense that only considers a small ball around an input sample. To address these threats, they provided to look at the decision boundaries around an example to characterize adversarial examples from the proposed OPTMARGIN, because the decision boundaries around them are different from that of normal examples.

Huang et al. [154] proposed a simple but effective model-agnostic detector based on the observation that the decision boundaries of the adversarial region are usually close to the legitimate instances. Yang et al. [151] found that adversarial attacks could lead to significant changes in feature attribution even if the visual perturbation were imperceptible. Therefore, they leveraged the feature attribution scores to distinguish the adversarial samples from clean examples. Given the impracticality of acquiring labeled instances for all possible adversarial attacks, Cintas et al. [155] proposed an unsupervised detection approach by means of a subset scanning technique commonly used in anomalous pattern detection. Ghosh et al. [156] proposed a **variational autoencoder (VAE)** model as a detector of adversarial samples. In this generative model, they tried to search for a latent variable to perform classification with a Gaussian mixture prior.

Discussion. Though it is impossible to distinguish adversarial examples and benign examples for human, some metrics might be influenced by adversarial perturbations for DNNs, such as the distances between the instance [152] and adjacent classes and dimensional properties [153]. Through these metrics of DNNs, the stealthiness of adversarial attacks would be destroyed, providing more possible solutions for defenses.

The defensive methods in different stages of the proposed framework are presented in Table 3, and a comparison of their performance is provided there. The performance presented includes the attack strength and complexity, which are mainly based on the performance against white-box attacks for the convenience of comparison. We note that strong defenses can defend against existing state-of-the-art attacks on most of datasets (e.g., PGD [3] or C&W [33]). So defenses based on verifying robustness can be considered as strong defensive methods. Moderate strength means that the defenses can defend against most existing attacks while being ineffective against strong attacks. And weak defenses represent the methods that aim to identify the malicious examples instead of providing satisfactory accuracy over these detected samples. In terms of complexity gauge, we note that the defenses that cannot be applied to large networks have high complexity. Defenses with moderate complexity can be scaled to large networks, but they still require additional training. Some efficient defenses without the requirements of additional training are deemed to be low complexity.

5 EXPLANATIONS FOR THE PHENOMENON OF ADVERSARIAL EXAMPLES

In this section, we discuss the existing hypotheses to provide a further understanding of adversarial examples against DNNs. These works analyzing the intrinsic vulnerabilities of DNNs fall under two headings: (1) Point of view of data and (2) Point of view of model.

5.1 Data Perspective

The training of DNNs usually demands a great deal of data with high quality to perform well, which is the driving force behind popular DNNs. However, the vulnerabilities of non-robust DNNs against adversarial examples can come from training data.

Table 3. Catalog of Adversarial Defenses Following the Analysis Framework Proposed and the Performance Demonstrated

Stage	Defenses	Defensive Strategy	Strength	Complexity	Experiment	
					Target Attack	**Dataset**
1	Two-layer model [117]	Over-approximation	Strong	High	PGD	M
1	ReLU-based model [118]	Over-approximation	Strong	High	F, PGD	M, FM, HAR, SVHN
1	Distributional robustness [119]	Over-approximation	Strong	Moderate	F, I-F, PGD	M
1	Co-design [120]	Over-approximation	Strong	Moderate	PGD	M, C-10
1	MIPVerify [121]	MILP	Strong	Moderate	PGD	M, C-10
1	RefineZono [122]	MILP	Strong	Moderate	L_∞norm Attack	M, C-10
2	FGSM-training [46]	Adversarial Training	Moderate	Moderate	F	M
2	PGD-training [3]	Adversarial Training	Strong	High	PGD	M, C-10
2	Triplet Loss [107]	Adversarial Training	Moderate	Moderate	I-F, C&W, DF	CvD, M, C-10
2	Discretization [123]	Adversarial Training	Strong	Moderate	F, PGD	M, C-10, C-100, SVHN
2	CAT [128]	Adversarial Training	Strong	Moderate	C&W, PGD	C-10, SVHN
2	Adv-BNN [124]	Adversarial Training	Strong	Moderate	PGD	C-10, STL-10, IN
2	EAT [126]	Adversarial Training	Moderate	Moderate	S-LL	M, IN
2	ADTA [130]	Adversarial Training	Strong	Moderate	F, PGD, Rand+F	FM, SVHN, C-10, C-100
2	Cascade AT [127]	Adversarial Training	Moderate	Moderate	S-LL, I-F, C&W	C-10
2	LAT [125]	Adversarial Training	Strong	Moderate	PGD	M, C-10, C-100, SVHN, IN
2	Distillation [131]	Distillation	Weak	Moderate	JSMA	M, C-10
2	ARD [132]	Distillation	Strong	Moderate	PGD	C-10, C-100
3	Controlling Lipschitz [108]	Architectural Defenses	Strong	Moderate	C&W	M, C-10
3	ABS [109]	Architectural Defenses	Strong	Moderate	C&W	M, C-10
3	Variance Layers [133]	Architectural Defenses	Moderate	Low	F	C-10
3	PixelDefend [110]	Input Transformation	Weak	Moderate	F, DF, C&W, BIM	FM, C-10
3	Defense-GAN [111]	Input Transformation	Strong	Moderate	F, R+F, C&W	M, FM
3	SAP [135]	Input Transformation	Moderate	Low	F	C-10
3	Randomization [136]	Input Transformation	Weak	Low	F, DF, C&W	IN
5	LID [153]	Detection	Weak	Low	F, BIM, JSMA	M, C-10, SVHN
5	Desion Boundary [152]	Detection	Weak	Low	F	M, C-10
5	Model-agnostic Detector [154]	Detection	Weak	Low	F, BIM, JSMA, DF, C&W	M, C-10, IN
5	Subset Scanning [155]	Detection	Weak	Low	F, BIM, DF	M, FM
5	VAE Detector [156]	Detection	Weak	Low	F	M, SVHN, COIL-100
5	ML-LOO [151]	Detection	Weak	Low	F, PGD, C&W	M, C-10, C-100

In the table, the following abbreviations have been used in the Dataset column: M for MNIST, FM for FashionMNIST, C for CIFAR, IN for ImageNet, CvD for Cats vs Dogs. In the Target Attack, the following abbreviations have been used: F for FGSM, S-LL for Single-Step Least-Likely Class Method, DF for DeepFool.

- **Non-robust features:** Ilyas et al. [55, 157] demonstrated that the phenomenon of adversarial examples is a consequence of data features. They split features into robust and non-robust features (incomprehensible to humans and more likely to be manipulated by attackers). Wang et al. [157] investigated the features extracted by DNNs from the perspective of frequency spectrum in the image domain. They observed high-frequency components are almost imperceptible to humans. Adversarial vulnerabilities can be considered as a consequence of generalization mysteries caused by non-robust high-frequency components.
- **High dimension:** To explore the relationship between data dimension and robustness, Gilmer et al. [158] induced a metric to quantify the robustness of classifiers. Specifically, X is denoted as the set of benign examples with label y. Given $x \in X$, x' is the nearest point with label $y' \neq y$, which is assigned to label y by target model. The average distance between x' and x can be used to quantify the robustness of target model. However, it is verified to be inversely proportional to the dimension of data d. Likewise, adversarial examples are shown to be inevitable for lots of problems, and high dimension of data could limit the robustness of models [159, 160].
- **Insufficient data:** Schmidt et al. [161] observed unavoidable adversarial examples are model-agnostic. Through some empirical results, they concluded that existing datasets are not large enough to obtain robust models. Hendrycks et al. [162] also proposed that pre-training on larger datasets can effectively improve the robustness, though the traditional classification performance is not enhanced.

5.2 Model Perspective

Alternatively, properties of DNNs were thought of as the potential cause of adversarial threats.

- **Non-linearity:** Szegedy et al. [2] first explored the adversarial vulnerability of DNNs from the perspective of models. They argued that high non-linearity of DNNs led to low probability pockets in the data manifold. These pockets can be found in the search progress of adversarial examples, though they are hard to reach randomly in the input space.
- **Linearity:** By contrast, Goodfellow et al. [46] refuted the non-linearity hypothesis and proposed that overlinearity of DNNs caused the vulnerability. Specifically, they thought some easy activation functions (e.g., ReLU and sigmoid) were likely to lead to linear behaviors of DNNs. As a result, summing small perturbations for high-dimensional inputs will amplify the perturbations and cause a misclassification. Fawzi et al. [163] also observed that linear classifiers are more prone to be misled by adversarial examples when compared to deeper models.
- **Decision boundary tilting:** Some researchers argued that local linearity behaviors alone cannot lead to adversarial vulnerability [164]. Tanay and Griffin [164] believed this might be caused by overfitted models. Moreover, they presented a concept, boundary tilting, to describe the phenomenon that the learned boundary of a well-trained model is close to the training data manifold but tilted beyond this manifold. As a result, perturbing the benign example towards the learned boundary can produce an effective adversarial example causing classification.
- **Training procedure:** Bubeck et al. [165] focused on the training procedure and considered adversarial vulnerability as an unavoidable result of computational constraints in standard training. Tsipras et al. [5] and Nakkiran et al. [166] also proposed that it is hard to attain accuracy and robustness simultaneously using current training techniques. More complex classifiers should be introduced for higher robustness.

5.3 Summary

Up to now, there is no unanimous explanation for the existence of adversarial examples for DNNs. Though several hypotheses have been proposed, some of them are even in conflict. Although some hypotheses are challenged as not convincing, there is still no sufficient evidence to deny them completely, because a number of attacks designed based on these hypotheses are verified empirically to be effective against DNNs. In our opinion, the vulnerability might be the joint effect of multiple hypotheses instead of one single property. As shown in Reference [164], complex data manifold can also lead to adversarial examples, implying linearity is not the only root cause. Therefore, to discover a unanimous hypothesis, it is necessary to bridge the inner connections between different factors. Specifically, factors from the perspective of data might be linked to the model-related hypotheses. For example, increasing the number of training data using data augmentation seemingly also helps mitigate the effect of tilting boundary from the perspective of model [166]. Moreover, adversarial training can force the DNNs to be less linear than counterparts using standard training [167], while it can also be explained as a class of methods for feature purification to remove the non-robust features [168]. Therefore, linking different hypotheses might be a direction to develop a universally accepted explanation for the existence of adversarial examples.

6 DATASETS

This section will provide a comprehensive introduction of the datasets used in adversarial learning. The **Attack Success Rate** (**ASR**, the proportion of adversarial examples achieving misclassification successfully) and adversarial accuracy (classification accuracy under adversarial examples) are two common metrics.

6.1 MNIST and CIFAR-10

The MNIST is a database of bi-level handwritten digits (from 0 to 9) with 60,000 training examples and 10,000 test examples. Each of them is translated to a 28×28 image via size normalization [169]. The CIFAR-10 contains 10 classes of color images with 32×32 pixels, where there are 50,000 and 10,000 images in training set and test set, respectively. Because of the simplicity and small size of MNIST and CIFAR-10, they have been proved to be easy to attack and defend. For example, as shown in Reference [170], when l_2 norm of adversarial perturbations reaches about 3 in white-box scenarios, some gradient-based methods (e.g., BIM, MI-FGSM, and PGD) can achieve ASR close to 100% for untargeted attacks on both MNIST and CIFAR-10.

6.2 CIFAR-100

CIFAR-100 is similar to the CIFAR-10 except it has 100 classes. For white-box untargeted attacks, both PGD and MI-FGSM can achieve high ASR on CIFAR-100, which is greater than 99% [123, 130]. However, it is harder to defend against adversarial examples. As shown in Reference [125], under the defense of standard adversarial training [3], PGD with 10 steps can still achieve the ASR of 77.28% on ResNet model [171].

6.3 SVHN

Similar to MNIST, SVHN is a dataset with 10 classes for 10 digits (from 0 to 9) [172]. SVHN contains real-world color images collected from house numbers in Google Street View images. There are 73,257 digits in training dataset and 26,032 images in test dataset. Buckman et al. [123] showed that when the l_∞ norm of perturbation is not greater than 0.047, PGD can reduce the adversarial accuracy on SVHN to 6.99%. However, white-box attacks on SVHN are also easy to defend, and the combination of adversarial training and discretization [123] can enable the model to achieve 94.77% adversarial accuracy.

6.4 ImageNet

ImageNet is a large-scale image dataset with over 14 million images, which has been instrumental in computer vision research [173]. In white-box settings, it is also easy to attack models with no defenses. For example, PGD with l_∞ distortion less than 0.01 can fool the VGG model with a probability of 100% [124]. More interestingly, the effectiveness of defenses against various attacks on ImageNet varies greatly [136], which are summarized in Table 4. Specifically, as shown in Table 4, both DeepFool and C&W can obtain 100% ASR for models with no defenses, while that of single-step FGSM is lower (66.8%). However, after adding additional randomization layers, the top-1 accuracy under FGSM is increased by 31.9% on Inception model [174]. The effects of iterative attacks (i.e., DeepFool and C&W) can be mitigated greatly through randomization mechanism, and the accuracy is increased by over 96%. This is caused by the over-fitting and weak transferability of iterative attacks. In addition, as shown in Table 4, defense model using adversarial training techniques can lead to satisfactory accuracy under single-step attacks, though adversarial training has little effect on iterative attacks.

In conclusion, it is easy for white-box attackers to achieve high ASR on these five commonly used datasets. However, the effectiveness of existing defenses against various attacks varies significantly, which leads to challenges of applying a single adversarial defense to eliminate threats from all adversarial attacks. Therefore, it will be a potential solution to combine multiple defenses, which adopt different strategies and are compatible with each other. However, the premise is that we must identify the differences between defensive methods and their compatibility. Therefore, we need to comprehensively consider existing methods as a whole and figure out the effect of

Table 4. Comparison between the Effectiveness of Different Attacks on ImageNet

Model	Inception		ResNet		Inception-ResNet		ens-adv-Inception-ResNet	
	target model	defense model	target model	defense model	target model	defense model	target model	defense model
FGSM	33.2%	65.1%	26.3%	71.8%	65.3%	81.0%	84.4%	95.7%
DeepFool	0%	98.3%	0%	97.7%	0%	98.2%	0.2%	99.1%
C&W	0%	96.9%	0%	97.1%	0.3%	97.7%	0.9%	98.8%

Ens-adv-Inception-ResNet is obtained by applying the ensemble adversarial training on Inception-ResNet. Target model represents naturally trained model with no defense. Defense model is the model with randomization-based defenses. Percentage is the Top-1 classification accuracy.

each type of method in the whole system. This is also our inspiration to propose the lifecycle for adversarial machine learning and allocate the existing methods to different stages.

7 FUTURE DIRECTIONS

Recently, **Machine Learning as-a-service (MLaaS)** has become a fashion, thanks to increases in data resources, computational capacity, and fundamental theory. In the future, deep learning systems will show increasing promise of being a mature integrative service in many areas, such as business, the military, the transportation, and even our daily lives. Unfortunately, according to our survey, the current deep learning systems in the real world are still far from perfect. Both development and deployment processes are still vulnerable to attack by malicious adversaries with the goal of stealing data, breaching privacy, or compromising the target model. To mitigate the safety and privacy concerns in deep learning and promote "deep learning as-a-service," there must be more studies on model security. So, this is a subject of discussion that is likely to remain active and vibrant for the very long-term. As such, a few possible future directions of research have been imagined here.

- ***Safety and Privacy Framework.*** As mentioned by Bae et al. [21], the research studies on both deep learning security and privacy are still fragmented, because the types of threats and their objectives are different. Secure deep learning aims for models with high robustness against malicious inputs. Alternatively, privacy-preserving deep learning aims to protect privacy of sensitive data of users involved in the training. In addition to the potential leakage of privacy associated with collaborative training, membership inference and model inversion attacks can cause threats to the privacy of users. The commonly used privacy-preserving techniques include Differential Privacy [175] and Cryptographic methods such as **Homomorphic Encryption (HE)** [176] and Multi-Party Computation (SMC) [177]. However, these methods have different strategies from the defenses in Adversarial Attacks that aim to mitigate the security threats. Significant previous works focusing on the analysis of privacy problems, including membership inference attacks [178, 179] and model inversion attacks, are an independent body literature from the study of model security in deep learning. And the relationship between the privacy issues and model security threats is still unclear. Therefore, it is very difficult at this juncture to propose a unifying analysis framework that addresses both privacy issues and security problems.

  In other words, a deep learning system that provides some kind of privacy guarantees may still have a low-level robustness, because privacy and security are two different types of threats that are still analyzed independently. Bae et al. [21] first proposed a notion of SPAI: Secure and Private AI. In this article, though we mainly focus on security threats in deep learning, we reviewed some papers aiming to resolve privacy issues, such as membership inference attacks [149, 150] in deep learning by crafting some adversarial examples. Song et al. [180] raise concerns over the influence of adversarial defenses on privacy leaks, because they found that a model robust to adversarial perturbations can be more sensitive to the training data. Thus, they explored the relationships between privacy and security threats.

If those relationships could be identified clearly today, then the frameworks proposed in this article could be adapted into a unifying analysis framework. Hence, one possible future direction of study would be to attempt to design methods that systematically train privacy-preserving and robust DNN models simultaneously.

- ***Adversarial Model Inversion Defenses.*** In the study of privacy threats studying, attackers can use deep learning models to implement membership inference and model inversion attacks. Differential Privacy techniques can effectively reduce the success rate of membership attacks [181]. However, there are few defenses against model inversion attacks. Further, the ones there are, such as Reference [182], require retraining, which means the solution comes at the cost of a high computational burden. Adversarial samples have emerged as countermeasures against membership inference attacks [149, 150], but the capacity of adversarial samples to defend against model inversion attacks has not been explored. Xiao et al. [183] proposed to borrow the idea of adversarial learning to train a privacy-preserving model against model inversion adversary. In the training phase, adversarial reconstruction loss is considered as a regularizer of the objective of the target model, which can decrease the similarity between the original image and the reconstructed image by an adversary. This fact makes implicit the possibility of using the adversarial attacks to explore the vulnerabilities of model inversion attack models designed to steal privacy. Specifically, malicious inversion models used to reconstruct images from the output of the target model can also be vulnerable to adversarial examples. And their training data, i.e., the output of the target model, are provided by the defenders. Because defenders have access to the training data of an inversion model, defenders could introduce a poisoning attack as a countermeasure to decrease the performance of inversion models. This would be an interesting direction to explore.
- ***Monitoring Methods.*** The defenses against APTs can be largely divided into three classes, including Monitoring Methods, Detection Methods, and Mitigation Methods. Monitoring methods can be regarded as one of most effective categories of defense. These approaches include Disk Monitoring, Log Monitoring, Code Monitoring, and so on. For example, an application's execution logs can produce a large amount of information, which can be used to design defenses. Bohara et al. [184] proposed an intrusion detection method based on four features extracted from the information in host logs. In addition, deep learning techniques provide effective methods for monitoring the disk and logs to detect the malicious behavior by an adversary or prevent attacks in the early stages. Du et al. [185] proposed a DeepLog neural network model based on the **Long Short-Term Memory (LSTM)** to automatically learn normal log patterns and detect anomalies. Inspired by the monitoring strategies in APT methods, we can also use the monitoring methods to study security problems in deep learning models. Specifically, the success of log monitoring models encourages us to explore the effectiveness of monitoring models on security threats of deep learning. Before training the target models, we can train a second model to analyze the information from logs. When we get a satisfying monitoring model, it can run with the training of our target model to monitor the whole training phase and determine whether malicious behaviors occur in the training, such as inserting poisoning examples. This strategy can detect the occurrence of suspicious examples through the deviation from normal log patterns, which can be a potential countermeasure against poisoning attacks occurring in the training phase and can prevent the deployment of compromised models.

8 CONCLUSIONS

Despite the incredible performance of DNNs for solving the tasks in our daily lives, the security problems of deep learning techniques have generally given rise to extensive concerns over how

vulnerable these models are to adversarial samples. A vast body of attacks and defensive mechanisms have been proposed to find adversarial samples to evaluate the security and robustness accurately.

In this survey, we first reviewed works related to adversarial attacks. We then proposed an analysis framework as an attempt to provide a standard evaluation process for adversarial attack threats. Inspired by the lifecycle of Advanced Persistent Threats, we mapped five stages of the life of an adversarial attack to the five stages of Alshamrani et al.'s [19] APT lifecycle, which can help understand these attack methods systematically. Moreover, we also provided a similar analysis framework with five stages for adversarial defenses. The objectives of defensive strategies in different stages correspond to that in the lifecycle of adversarial attacks. Under our proposed framework, one can combine multiple types of defenses in various stages to minimize the risks to the target models. The survey concludes with a discussion on possible fruitful directions of future study to improve existing adversarial attacks and defenses.

REFERENCES

[1] Yue Zhao, Hong Zhu, Ruigang Liang, Qintao Shen, Shengzhi Zhang, and Kai Chen. 2019. Seeing isn't believing: Towards more robust adversarial attack against real world object detectors. In *Proceedings of theACM SIGSAC Conference on Computer and Communications Security*.

[2] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *Proceedings of the 2nd International Conference on Learning Representations*.

[3] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *Proceedings of the 6th International Conference on Learning Representations*.

[4] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. 2017. Adversarial examples in the physical world. In *Proceedings of the 5th International Conference on Learning Representations*.

[5] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. 2019. Robustness may be at odds with accuracy. In *Proceedings of the 7th International Conference on Learning Representations*.

[6] Hengtong Zhang, Tianhang Zheng, Jing Gao, Chenglin Miao, Lu Su, Yaliang Li, and Kui Ren. 2019. Data poisoning attack against knowledge graph embedding. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*.

[7] Yuzhe Ma, Xiaojin Zhu, and Justin Hsu. 2019. Data poisoning against differentially-private learners: Attacks and defenses. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*.

[8] Ambra Demontis, Marco Melis, Maura Pintor, Matthew Jagielski, Battista Biggio, Alina Oprea, Cristina Nita-Rotaru, and Fabio Roli. 2019. Why do adversarial attacks transfer? Explaining transferability of evasion and poisoning attacks. In *Proceedings of the 28th USENIX Security Symposium*. 321–338.

[9] China Electronics Standardization Institute. 2021. *Artificial Intelligence Standardization White Paper*. Retrieved from http://www.cesi.cn/202107/7795.html.

[10] ISO/IEC 22989. 2021. *Information Technology - Artificial Intelligence - Artificial Intelligence Concepts and Terminology*. Retrieved from https://www.iso.org/obp/ui/#iso:std:iso-iec:22989:dis:ed-1:v1:en.

[11] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. 2021. A survey on adversarial attacks and defences. *CAAI Trans. Intell. Technol.* 6, 1 (2021), 25–45.

[12] Yupeng Hu, Wenxin Kuang, Zheng Qin, Kenli Li, Jiliang Zhang, Yansong Gao, Wenjia Li, and Keqin Li. 2021. Artificial intelligence security: Threats and countermeasures. *ACM Comput. Surv.* 55, 1 (2021), 1–36.

[13] Alexandru Constantin Serban, Erik Poll, and Joost Visser. 2020. Adversarial examples on object recognition: A comprehensive survey. *ACM Comput. Surv.* 53, 3 (2020), 66:1–66:38. DOI: http://dx.doi.org/10.1145/3398394

[14] 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 (2021), 1–38.

[15] Xingwei Zhang, Xiaolong Zheng, and Wenji Mao. 2021. Adversarial perturbation defense on deep neural networks. *ACM Comput. Surv.* 54, 8 (2021), 1–36.

[16] Yashar Deldjoo, Tommaso Di Noia, and Felice Antonio Merra. 2021. A survey on adversarial recommender systems: from attack/defense strategies to generative adversarial networks. *ACM Comput. Surv.* 54, 2 (2021), 1–38.

[17] Ishai Rosenberg, Asaf Shabtai, Yuval Elovici, and Lior Rokach. 2021. Adversarial machine learning attacks and defense methods in the cyber security domain. *ACM Comput. Surv.* 54, 5 (2021), 1–36.

[18] Samuel G. Finlayson, John D. Bowers, Joichi Ito, Jonathan L. Zittrain, Andrew L. Beam, and Isaac S. Kohane. 2019. Adversarial attacks on medical machine learning. *Science* 363, 6433 (2019), 1287–1289.

[19] Adel Alshamrani, Sowmya Myneni, Ankur Chowdhary, and Dijiang Huang. 2019. A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities. *IEEE Commun. Surv. Tutor.* 21, 2 (2019), 1851–1877.

[20] Han Xu, Yao Ma, Haochen Liu, Debayan Deb, Hui Liu, Jiliang Tang, and Anil K. Jain. 2020. Adversarial attacks and defenses in images, graphs and text: A review. *Int. J. Autom. Comput.* 17, 2 (2020), 151–178. DOI: http://dx.doi.org/10.1007/s11633-019-1211-x

[21] Ho Bae, Jaehee Jang, Dahuin Jung, Hyemi Jang, Heonseok Ha, and Sungroh Yoon. 2018. Security and privacy issues in deep learning. *CoRR* abs/1807.11655 (2018).

[22] Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu. 2020. Adversarial attacks and defenses in deep learning. *Engineering* 6, 3 (2020), 346–360. DOI: http://dx.doi.org/10.1016/j.eng.2019.12.012

[23] Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. 2020. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *arXiv preprint arXiv:2012.10544* (2020).

[24] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C. Lupu, and Fabio Roli. 2017. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. 27–38.

[25] Chaofei Yang, Qing Wu, Hai Li, and Yiran Chen. 2017. Generative poisoning attack method against neural networks. *arXiv preprint arXiv:1703.01340* (2017).

[26] W. Ronny Huang, Jonas Geiping, Liam Fowl, Gavin Taylor, and Tom Goldstein. 2020. Metapoison: Practical general-purpose clean-label data poisoning. *Adv. Neural Inf. Process. Syst.* 33 (2020), 12080–12091.

[27] Xiaoyu Yang, Yuefei Lyu, Tian Tian, Yifei Liu, Yudong Liu, and Xi Zhang. 2020. Rumor detection on social media with graph structured adversarial learning. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*.

[28] Xingxing Wei, Siyuan Liang, Ning Chen, and Xiaochun Cao. 2019. Transferable adversarial attacks for image and video object detection. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*.

[29] Chenxiao Zhao, P. Thomas Fletcher, Mixue Yu, Yaxin Peng, Guixu Zhang, and Chaomin Shen. 2019. The adversarial attack and detection under the Fisher information metric. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*.

[30] 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*.

[31] Zhipeng Wei, Jingjing Chen, Xingxing Wei, Linxi Jiang, Tat-Seng Chua, Fengfeng Zhou, and Yu-Gang Jiang. 2020. Heuristic black-box adversarial attacks on video recognition models. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*.

[32] Yang Zhang, Hassan Foroosh, Philip David, and Boqing Gong. 2019. CAMOU: Learning physical vehicle camouflages to adversarially attack detectors in the wild. In *Proceedings of the 7th International Conference on Learning Representations*.

[33] Nicholas Carlini and David A. Wagner. 2017. Towards evaluating the robustness of neural networks. In *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE Computer Society, 39–57. DOI: http://dx.doi.org/10.1109/SP.2017.49

[34] Nicolas Papernot, Patrick D. 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*.

[35] Yulong Cao, Chaowei Xiao, Benjamin Cyr, Yimeng Zhou, Won Park, Sara Rampazzi, Qi Alfred Chen, Kevin Fu, and Z. Morley Mao. 2019. Adversarial sensor attack on LiDAR-based perception in autonomous driving. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*.

[36] Zhaohui Che, Ali Borji, Guangtao Zhai, Suiyi Ling, Jing Li, and Patrick Le Callet. 2020. A new ensemble adversarial attack powered by long-term gradient memories. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*. 3405–3413.

[37] Rima Alaifari, Giovanni S. Alberti, and Tandri Gauksson. 2019. ADef: An iterative algorithm to construct adversarial deformations. In *Proceedings of the 7th International Conference on Learning Representations*.

[38] Wieland Brendel, Jonas Rauber, and Matthias Bethge. 2018. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *Proceedings of the 6th International Conference on Learning Representations*.

[39] Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the ACM on Asia Conference on Computer and Communications Security*.

[40] Ali Shafahi, W. Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. 2018. Poison frogs! Targeted clean-label poisoning attacks on neural networks. *Adv. Neural Inf. Process. Syst.* 31 (2018).
[41] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. 2017. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526* (2017).
[42] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. 2020. Hidden trigger backdoor attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 11957–11965.
[43] Akhilan Boopathy, Sijia Liu, Gaoyuan Zhang, Cynthia Liu, Pin-Yu Chen, Shiyu Chang, and Luca Daniel. 2020. Proper network interpretability helps adversarial robustness in classification. In *Proceedings of the International Conference on Machine Learning*. PMLR, 1014–1023.
[44] Gavin Weiguang Ding, Kry Yik Chau Lui, Xiaomeng Jin, Luyu Wang, and Ruitong Huang. 2019. On the sensitivity of adversarial robustness to input data distributions. In *Proceedings of the 7th International Conference on Learning Representations*.
[45] 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 33rd AAAI Conference on Artificial Intelligence*.
[46] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *Proceedings of the 3rd International Conference on Learning Representations*.
[47] Kaidi Xu, Hongge Chen, Sijia Liu, Pin-Yu Chen, Tsui-Wei Weng, Mingyi Hong, and Xue Lin. 2019. Topology attack and defense for graph neural networks: An optimization perspective. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 3961–3967.
[48] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2017. Universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 86–94.
[49] Heng Chang, Yu Rong, Tingyang Xu, Wenbing Huang, Honglei Zhang, Peng Cui, Wenwu Zhu, and Junzhou Huang. 2020. A restricted black-box adversarial framework towards attacking graph embedding models. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*.
[50] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramèr, Atul Prakash, and Tadayoshi Kohno. 2018. Physical adversarial examples for object detectors. In *Proceedings of the 12th USENIX Workshop on Offensive Technologies*.
[51] Hiromu Yakura and Jun Sakuma. 2019. Robust audio adversarial example for a physical attack. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 5334–5341.
[52] Kaidi Xu, Sijia Liu, Pu Zhao, Pin-Yu Chen, Huan Zhang, Quanfu Fan, Deniz Erdogmus, Yanzhi Wang, and Xue Lin. 2019. Structured adversarial attack: Towards general implementation and better interpretability. In *Proceedings of the 7th International Conference on Learning Representations*.
[53] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. 2018. Spatially transformed adversarial examples. In *Proceedings of the 6th International Conference on Learning Representations*.
[54] Hsueh-Ti Derek Liu, Michael Tao, Chun-Liang Li, Derek Nowrouzezahrai, and Alec Jacobson. 2019. Beyond pixel norm-balls: Parametric adversaries using an analytically differentiable renderer. In *Proceedings of the 7th International Conference on Learning Representations*.
[55] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. 2019. Adversarial examples are not bugs, they are features. In *Proceedings of the Annual Conference on Neural Information Processing Systems*.
[56] 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*.
[57] Huan Zhang, Hongge Chen, Zhao Song, Duane S. Boning, Inderjit S. Dhillon, and Cho-Jui Hsieh. 2019. The limitations of adversarial training and the blind-spot attack. In *Proceedings of the 7th International Conference on Learning Representations*.
[58] 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.
[59] Jinghui Chen, Dongruo Zhou, Jinfeng Yi, and Quanquan Gu. 2020. A Frank-Wolfe framework for efficient and effective adversarial attacks. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*.
[60] Tianhang Zheng, Changyou Chen, and Kui Ren. 2019. Distributionally adversarial attack. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*.
[61] Francesco Croce and Matthias Hein. 2020. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proceedings of the International Conference on Machine Learning*. PMLR, 2206–2216.
[62] Francesco Croce and Matthias Hein. 2021. Mind the box: l_1-APGD for sparse adversarial attacks on image classifiers. In *Proceedings of the International Conference on Machine Learning*. PMLR, 2201–2211.
[63] Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. In *Proceedings of the 6th International Conference on Learning Representations*.

[64] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. 2018. Generating adversarial examples with adversarial networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*.
[65] Huy Phan, Yi Xie, Siyu Liao, Jie Chen, and Bo Yuan. 2020. CAG: A real-time low-cost enhanced-robustness high-transferability content-aware adversarial attack generator. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*.
[66] Sayantan Sarkar, Ankan Bansal, Upal Mahbub, and Rama Chellappa. 2017. UPSET and ANGRI: Breaking high performance image classifiers. *arXiv preprint arXiv:1707.01159* (2017).
[67] Ali Shafahi, Mahyar Najibi, Zheng Xu, John P. Dickerson, Larry S. Davis, and Tom Goldstein. 2020. Universal adversarial training. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*.
[68] Kenneth T. Co, Luis Muñoz-González, Sixte de Maupeou, and Emil C. Lupu. 2019. Procedural noise adversarial examples for black-box attacks on deep convolutional networks. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. 275–289.
[69] Chaoning Zhang, Philipp Benz, Chenguo Lin, Adil Karjauv, Jing Wu, and In So Kweon. 2021. A survey on universal adversarial attack. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*. ijcai.org, 4687–4694. DOI: http://dx.doi.org/10.24963/ijcai.2021/635
[70] Yash Sharma, Gavin Weiguang Ding, and Marcus A. Brubaker. 2019. On the effectiveness of low frequency perturbations. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 3389–3396.
[71] Zheng Yuan, Jie Zhang, Yunpei Jia, Chuanqi Tan, Tao Xue, and Shiguang Shan. 2021. Meta gradient adversarial attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7748–7757.
[72] Xiaosen Wang and Kun He. 2021. Enhancing the transferability of adversarial attacks through variance tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1924–1933.
[73] Zhibo Wang, Hengchang Guo, Zhifei Zhang, Wenxin Liu, Zhan Qin, and Kui Ren. 2021. Feature importance-aware transferable adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7639–7648.
[74] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 2017. Zoo: 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*.
[75] Chun-Chen Tu, Pai-Shun 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 33rd AAAI Conference on Artificial Intelligence*.
[76] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. 2018. Black-box adversarial attacks with limited queries and information. In *Proceedings of the International Conference on Machine Learning*. PMLR, 2137–2146.
[77] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. 2020. Square attack: A query-efficient black-box adversarial attack via random search. In *Proceedings of the European Conference on Computer Vision*. Springer, 484–501.
[78] Maksym Yatsura, Jan Metzen, and Matthias Hein. 2021. Meta-learning the search distribution of black-box random search based adversarial attacks. *Advances in Neural Information Processing Systems* 34 (2021).
[79] Minhao Cheng, Thong Le, Pin-Yu Chen, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. 2019. Query-efficient hard-label black-box attack: An optimization-based approach. In *Proceedings of the 7th International Conference on Learning Representations*.
[80] Minhao Cheng, Simranjit Singh, Patrick Chen, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. 2019. Sign-opt: A query-efficient hard-label adversarial attack. *arXiv preprint arXiv:1909.10773* (2019).
[81] Thibault Maho, Teddy Furon, and Erwan Le Merrer. 2021. SurFree: A fast surrogate-free black-box attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10430–10439.
[82] Andrew Ilyas, Logan Engstrom, and Aleksander Madry. 2019. Prior convictions: Black-box adversarial attacks with bandits and priors. In *Proceedings of the 7th International Conference on Learning Representations*.
[83] Nina Narodytska and Shiva Prasad Kasiviswanathan. 2016. Simple black-box adversarial perturbations for deep networks. *arXiv preprint arXiv:1612.06299* (2016).
[84] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. 2019. One pixel attack for fooling deep neural networks. *IEEE Trans. Evolut. Computat.* 23, 5 (2019), 828–841.
[85] Yan Feng, Bin Chen, Tao Dai, and Shu-Tao Xia. 2020. Adversarial attack on deep product quantization network for image retrieval. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*.
[86] Tzungyu Tsai, Kaichen Yang, Tsung-Yi Ho, and Yier Jin. 2020. Robust adversarial objects against deep learning models. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*.
[87] Elias B. Khalil, Amrita Gupta, and Bistra Dilkina. 2019. Combinatorial attacks on binarized neural networks. In *Proceedings of the 7th International Conference on Learning Representations*.
[88] Anshuman Chhabra, Abhishek Roy, and Prasant Mohapatra. 2020. Suspicion-free adversarial attacks on clustering algorithms. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*.

[89] Dayong Ye, Minjie Zhang, and Danny Sutanto. 2014. Cloning, resource exchange, and relationadaptation: An integrative self-organisation mechanism in a distributed agent network. *IEEE Trans. Parallel Distrib. Syst.* 25, 4 (2014), 887–897. DOI: http://dx.doi.org/10.1109/TPDS.2013.120

[90] Dayong Ye and Minjie Zhang. 2015. A self-adaptive strategy for evolution of cooperation in distributed networks. *IEEE Trans. Comput.* 64, 4 (2015), 899–911. DOI: http://dx.doi.org/10.1109/TC.2014.2308188

[91] Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. 2017. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284* (2017).

[92] Xian Wu, Wenbo Guo, Hua Wei, and Xinyu Xing. 2021. Adversarial policy training against deep reinforcement learning. In *Proceedings of the 30th USENIX Security Symposium (USENIX Security'21)*. 1883–1900.

[93] Shuyu Cheng, Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. 2019. Improving black-box adversarial attacks with a transfer-based prior. *Adv. Neural Inf. Process. Syst.* 32 (2019).

[94] Giulio Lovisotto, Henry Turner, Ivo Sluganovic, Martin Strohmeier, and Ivan Martinovic. 2021. SLAP: Improving physical adversarial examples with short-lived adversarial perturbations. In *Proceedings of the 30th USENIX Security Symposium (USENIX Security'21)*.

[95] Abdullah Hamdi, Matthias Mueller, and Bernard Ghanem. 2020. SADA: Semantic adversarial diagnostic attacks for autonomous applications. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*.

[96] Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2018. Deep text classification can be fooled. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*.

[97] Erwin Quiring, Alwin Maier, and Konrad Rieck. 2019. Misleading authorship attribution of source code using adversarial learning. In *Proceedings of the 28th USENIX Security Symposium*.

[98] Huangzhao Zhang, Zhuo Li, Ge Li, Lei Ma, Yang Liu, and Zhi Jin. 2020. Generating adversarial examples for holding robustness of source code processing models. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*.

[99] Xiaolei Liu, Kun Wan, Yufei Ding, Xiaosong Zhang, and Qingxin Zhu. 2020. Weighted-sampling audio adversarial example attack. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*.

[100] Hongting Zhang, Pan Zhou, Qiben Yan, and Xiao-Yang Liu. 2020. Generating robust audio adversarial examples with temporal dependency. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*.

[101] Xingxing Wei, Jun Zhu, Sha Yuan, and Hang Su. 2019. Sparse adversarial perturbations for videos. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*.

[102] Yuan Gong, Boyang Li, Christian Poellabauer, and Yiyu Shi. 2019. Real-time adversarial attacks. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 4672–4680.

[103] Moustapha M. Cisse, Yossi Adi, Natalia Neverova, and Joseph Keshet. 2017. Houdini: Fooling deep structured visual and speech recognition models with adversarial examples. *Adv. Neural Inf. Process. Syst.* 30 (2017).

[104] Pete Warden. 2018. Speech commands: A dataset for limited-vocabulary speech recognition. *CoRR* abs/1804.03209 (2018).

[105] Tsui Wei, Huan Zhang, Pin Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho Jui Hsieh, and Luca Daniel. 2018. Evaluating the robustness of neural networks: An extreme value theory approach. In *Proceedings of the 6th International Conference on Learning Representations*.

[106] Wenjie Ruan, Min Wu, Youcheng Sun, Xiaowei Huang, Daniel Kroening, and Marta Kwiatkowska. 2019. Global robustness evaluation of deep neural networks with provable guarantees for the Hamming distance. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 5944–5952.

[107] Pengcheng Li, Jinfeng Yi, Bowen Zhou, and Lijun Zhang. 2019. Improving the robustness of deep neural networks via adversarial training with triplet loss. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*.

[108] Haifeng Qian and Mark N. Wegman. 2019. L2-nonexpansive neural networks. In *Proceedings of the 7th International Conference on Learning Representations*.

[109] Lukas Schott, Jonas Rauber, Matthias Bethge, and Wieland Brendel. 2019. Towards the first adversarially robust neural network model on MNIST. In *Proceedings of the 7th International Conference on Learning Representations*.

[110] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. 2018. PixelDefend: Leveraging generative models to understand and defend against adversarial examples. In *Proceedings of the 6th International Conference on Learning Representations*.

[111] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. 2018. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *Proceedings of the 6th International Conference on Learning Representations*.

[112] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. 2018. Countering adversarial images using input transformations. In *Proceedings of the 6th International Conference on Learning Representations*.

[113] Gaurav Goswami, Nalini K. Ratha, Akshay Agarwal, Richa Singh, and Mayank Vatsa. 2018. Unravelling robustness of deep learning based face recognition against adversarial attacks. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

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

[115] Ian Goodfellow. 2018. Gradient masking causes clever to overestimate adversarial perturbation size. *arXiv preprint arXiv:1804.07870* (2018).
[116] Fuxun Yu, Zhuwei Qin, Chenchen Liu, Liang Zhao, Yanzhi Wang, and Xiang Chen. 2019. Interpreting and evaluating neural network robustness. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*.
[117] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. 2018. Certified defenses against adversarial examples. In *Proceedings of the 6th International Conference on Learning Representations*.
[118] Eric Wong and J. Zico Kolter. 2018. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *Proceedings of the 35th International Conference on Machine Learning*.
[119] Aman Sinha, Hongseok Namkoong, and John C. Duchi. 2018. Certifying some distributional robustness with principled adversarial training. In *Proceedings of the 6th International Conference on Learning Representations*.
[120] Kai Y. Xiao, Vincent Tjeng, Nur Muhammad (Mahi) Shafiullah, and Aleksander Madry. 2019. Training for faster adversarial robustness verification via inducing ReLU stability. In *Proceedings of the 7th International Conference on Learning Representations*.
[121] Vincent Tjeng, Kai Y. Xiao, and Russ Tedrake. 2019. Evaluating robustness of neural networks with mixed integer programming. In *Proceedings of the 7th International Conference on Learning Representations*.
[122] Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin T. Vechev. 2019. Boosting robustness certification of neural networks. In *Proceedings of the 7th International Conference on Learning Representations*.
[123] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian J. Goodfellow. 2018. Thermometer encoding: One hot way to resist adversarial examples. In *Proceedings of the 6th International Conference on Learning Representations*.
[124] Xuanqing Liu, Yao Li, Chongruo Wu, and Cho-Jui Hsieh. 2019. Adv-BNN: Improved adversarial defense through robust Bayesian neural network. In *Proceedings of the 7th International Conference on Learning Representations*.
[125] Nupur Kumari, Mayank Singh, Abhishek Sinha, Harshitha Machiraju, Balaji Krishnamurthy, and Vineeth N. Balasubramanian. 2019. Harnessing the vulnerability of latent layers in adversarially trained models. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 2779–2785.
[126] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian J. Goodfellow, Dan Boneh, and Patrick D. McDaniel. 2018. Ensemble adversarial training: Attacks and defenses. In *Proceedings of the 6th International Conference on Learning Representations*.
[127] Taesik Na, Jong Hwan Ko, and Saibal Mukhopadhyay. 2018. Cascade adversarial machine learning regularized with a unified embedding. In *Proceedings of the 6th International Conference on Learning Representations*.
[128] Qi-Zhi Cai, Chang Liu, and Dawn Song. 2018. Curriculum adversarial training. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 3740–3747.
[129] Farzan Farnia, Jesse M. Zhang, and David Tse. 2019. Generalizable adversarial training via spectral normalization. In *Proceedings of the 7th International Conference on Learning Representations*.
[130] Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. 2019. Improving the generalization of adversarial training with domain adaptation. In *Proceedings of the 7th International Conference on Learning Representations*.
[131] Nicolas Papernot, Patrick D. 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*.
[132] Micah Goldblum, Liam Fowl, Soheil Feizi, and Tom Goldstein. 2020. Adversarially robust distillation. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*.
[133] Kirill Neklyudov, Dmitry Molchanov, Arsenii Ashukha, and Dmitry P. Vetrov. 2019. Variance networks: When expectation does not meet your expectations. In *Proceedings of the 7th International Conference on Learning Representations*.
[134] Jörn-Henrik Jacobsen, Jens Behrmann, Richard S. Zemel, and Matthias Bethge. 2019. Excessive invariance causes adversarial vulnerability. In *Proceedings of the 7th International Conference on Learning Representations*.
[135] Guneet S. Dhillon, Kamyar Azizzadenesheli, Zachary C. Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Animashree Anandkumar. 2018. Stochastic activation pruning for robust adversarial defense. In *Proceedings of the 6th International Conference on Learning Representations*.
[136] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan L. Yuille. 2018. Mitigating adversarial effects through randomization. In *Proceedings of the 6th International Conference on Learning Representations*.
[137] Chong Xiang, Arjun Nitin Bhagoji, Vikash Sehwag, and Prateek Mittal. 2021. PatchGuard: A provably robust defense against adversarial patches via small receptive fields and masking. In *Proceedings of the 30th USENIX Security Symposium (USENIX Security'21)*.
[138] Zhuolin Yang, Bo Li, Pin-Yu Chen, and Dawn Song. 2019. Characterizing audio adversarial examples using temporal dependency. In *Proceedings of the 7th International Conference on Learning Representations*.
[139] Shehzeen Hussain, Paarth Neekhara, Shlomo Dubnov, Julian McAuley, and Farinaz Koushanfar. 2021. WaveGuard: Understanding and mitigating audio adversarial examples. In *Proceedings of the 30th USENIX Security Symposium (USENIX Security'21)*.

[140] Jan Svoboda, Jonathan Masci, Federico Monti, Michael M. Bronstein, and Leonidas J. Guibas. 2019. PeerNets: Exploiting peer wisdom against adversarial attacks. In *Proceedings of the 7th International Conference on Learning Representations*.

[141] Huijun Wu, Chen Wang, Yuriy Tyshetskiy, Andrew Docherty, Kai Lu, and Liming Zhu. 2019. Adversarial examples for graph data: Deep insights into attack and defense. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 4816–4823.

[142] Adam Goodge, Bryan Hooi, See-Kiong Ng, and Wee Siong Ng. 2020. Robustness of autoencoders for anomaly detection under adversarial impact. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*.

[143] Dayong Ye, Tianqing Zhu, Sheng Shen, Wanlei Zhou, and Philip Yu. 2020. Differentially private multi-agent planning for logistic-like problems. *IEEE Trans. Depend. Secure Comput.* 19, 2 (2020), 1212–1226.

[144] Dayong Ye, Tianqing Zhu, Zishuo Cheng, Wanlei Zhou, and S. Yu Philip. 2020. Differential advising in multiagent reinforcement learning. *IEEE Trans. Cybern.* 52, 6 (2020), 5508–5521.

[145] Tao Zhang, Tianqing Zhu, Ping Xiong, Huan Huo, Zahir Tari, and Wanlei Zhou. 2020. Correlated differential privacy: Feature selection in machine learning. *IEEE Trans. Industr. Inform.* 16, 3 (2020), 2115–2124. DOI: http://dx.doi.org/10.1109/TII.2019.2936825

[146] Tianqing Zhu, Dayong Ye, Wei Wang, Wanlei Zhou, and Philip Yu. 2020. More than privacy: Applying differential privacy in key areas of artificial intelligence. *IEEE Trans. Knowl. Data Eng.* (2020), 1–1. DOI: http://dx.doi.org/10.1109/TKDE.2020.3014246

[147] Lefeng Zhang, Tianqing Zhu, Ping Xiong, Wanlei Zhou, and Philip S. Yu. 2021. More than privacy: Adopting differential privacy in game-theoretic mechanism design. *ACM Comput. Surv.* 54, 7 (July 2021). DOI: http://dx.doi.org/10.1145/3460771

[148] Dayong Ye, Tianqing Zhu, Sheng Shen, and Wanlei Zhou. 2020. A differentially private game theoretic approach for deceiving cyber adversaries. *IEEE Trans. Inf. Forens. Secur.* 16 (2020), 569–584.

[149] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. 2019. MemGuard: Defending against black-box membership inference attacks via adversarial examples. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. 259–274.

[150] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2018. Machine learning with membership privacy using adversarial regularization. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*.

[151] Puyudi Yang, Jianbo Chen, Cho-Jui Hsieh, Jane-Ling Wang, and Michael I. Jordan. 2020. ML-LOO: Detecting adversarial examples with feature attribution. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*.

[152] Warren He, Bo Li, and Dawn Song. 2018. Decision boundary analysis of adversarial examples. In *Proceedings of the 6th International Conference on Learning Representations*.

[153] Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi N. R. Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E. Houle, and James Bailey. 2018. Characterizing adversarial subspaces using local intrinsic dimensionality. In *Proceedings of the 6th International Conference on Learning Representations*.

[154] Bo Huang, Yi Wang, and Wei Wang. 2019. Model-agnostic adversarial detection by random perturbations. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*.

[155] Celia Cintas, Skyler Speakman, Victor Akinwande, William Ogallo, Komminist Weldemariam, Srihari Sridharan, and Edward McFowland. 2020. Detecting adversarial attacks via subset scanning of autoencoder activations and reconstruction error. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*. 876–882.

[156] Partha Ghosh, Arpan Losalka, and Michael J. Black. 2019. Resisting adversarial attacks using Gaussian mixture variational autoencoders. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*.

[157] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P. Xing. 2020. High-frequency component helps explain the generalization of convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8684–8694.

[158] Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S. Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. 2018. The relationship between high-dimensional geometry and adversarial examples. *arXiv preprint arXiv:1801.02774* (2018).

[159] Ali Shafahi, W. Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. 2018. Are adversarial examples inevitable? *arXiv preprint arXiv:1809.02104* (2018).

[160] Alhussein Fawzi, Hamza Fawzi, and Omar Fawzi. 2018. Adversarial vulnerability for any classifier. *Adv. Neural Inf. Process. Syst.* 31 (2018).

[161] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. 2018. Adversarially robust generalization requires more data. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 5019–5031.

[162] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. 2019. Using pre-training can improve model robustness and uncertainty. In *Proceedings of the International Conference on Machine Learning*. PMLR, 2712–2721.

[163] Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2015. Fundamental limits on adversarial robustness. In *Proceedings of the ICML, Workshop on Deep Learning*.
[164] Thomas Tanay and Lewis Griffin. 2016. A boundary tilting persepective on the phenomenon of adversarial examples. *arXiv preprint arXiv:1608.07690* (2016).
[165] Sébastien Bubeck, Yin Tat Lee, Eric Price, and Ilya Razenshteyn. 2019. Adversarial examples from computational constraints. In *Proceedings of the International Conference on Machine Learning*. PMLR, 831–840.
[166] Preetum Nakkiran. 2019. Adversarial robustness may be at odds with simplicity. *arXiv preprint arXiv:1901.00532* (2019).
[167] Adnan Siraj Rakin, Zhezhi He, Boqing Gong, and Deliang Fan. 2018. Blind pre-processing: A robust defense method against adversarial examples. *arXiv preprint arXiv:1802.01549* (2018).
[168] Zeyuan Allen-Zhu and Yuanzhi Li. 2022. Feature purification: How adversarial training performs robust deep learning. In *Proceedings of the IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 977–988.
[169] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.
[170] Jing Wu, Mingyi Zhou, Ce Zhu, Yipeng Liu, Mehrtash Harandi, and Li Li. 2021. Performance evaluation of adversarial attacks: Discrepancies and solutions. *arXiv preprint arXiv:2104.11103* (2021).
[171] 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.
[172] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*.
[173] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein et al. 2015. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* 115, 3 (2015), 211–252.
[174] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2818–2826.
[175] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. 2018. Scalable private learning with PATE. In *Proceedings of the 6th International Conference on Learning Representations*. OpenReview.net.
[176] Amartya Sanyal, Matt J. Kusner, Adrià Gascón, and Varun Kanade. 2018. TAPAS: Tricks to accelerate (encrypted) prediction as a service. In *Proceedings of the 35th International Conference on Machine Learning*.
[177] Bita Darvish Rouhani, M. Sadegh Riazi, and Farinaz Koushanfar. 2018. DeepSecure: Scalable provably-secure deep learning. In *Proceedings of the 55th Annual Design Automation Conference*. ACM, 2:1–2:6.
[178] 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)*. 3–18. DOI: http://dx.doi.org/10.1109/SP.2017.41
[179] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *Proceedings of the IEEE Symposium on Security and Privacy (SP)*. 739–753.
[180] Liwei Song, Reza Shokri, and Prateek Mittal. 2019. Privacy risks of securing machine learning models against adversarial examples. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*.
[181] Michael Backes, Pascal Berrang, Mathias Humbert, and Praveen Manoharan. 2016. Membership privacy in MicroRNA-based studies. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. ACM.
[182] Tianhao Wang, Yuheng Zhang, and Ruoxi Jia. 2021. Improving robustness to model inversion attacks via mutual information regularization. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*. AAAI Press, 11666–11673.
[183] Taihong Xiao, Yi-Hsuan Tsai, Kihyuk Sohn, Manmohan Chandraker, and Ming-Hsuan Yang. 2020. Adversarial learning of privacy-preserving and task-oriented representations. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*.
[184] Atul Bohara, Uttam Thakore, and William H. Sanders. 2016. Intrusion detection in enterprise systems by combining and clustering diverse monitor data. In *Proceedings of the Symposium and Bootcamp on the Science of Security*. ACM.
[185] Min Du, Feifei Li, Guineng Zheng, and Vivek Srikumar. 2017. *DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning*. Association for Computing Machinery, New York, NY, 1285–1298. DOI: https://doi.org/10.1145/3133956.3134015

Received 10 October 2021; revised 19 May 2022; accepted 4 July 2022