Artificial Intelligence Security: Threats and Countermeasures

YUPENG HU, WENXIN KUANG, ZHENG QIN, KENLI LI, and JILIANG ZHANG,
Hunan University
YANSONG GAO, Nanjing University of Science and Technology
WENJIA LI, New York Institute of Technology
KEQIN LI, State University of New York

In recent years, with rapid technological advancement in both computing hardware and algorithm, Artificial Intelligence (AI) has demonstrated significant advantage over human being in a wide range of fields, such as image recognition, education, autonomous vehicles, finance, and medical diagnosis. However, AI-based systems are generally vulnerable to various security threats throughout the whole process, ranging from the initial data collection and preparation to the training, inference, and final deployment. In an AI-based system, the data collection and pre-processing phase are vulnerable to sensor spoofing attacks and scaling attacks, respectively, while the training and inference phases of the model are subject to poisoning attacks and adversarial attacks, respectively. To address these severe security threats against the AI-based systems, in this article, we review the challenges and recent research advances for security issues in AI, so as to depict an overall blueprint for AI security. More specifically, we first take the lifecycle of an AI-based system as a guide to introduce the security threats that emerge at each stage, which is followed by a detailed summary for corresponding countermeasures. Finally, some of the future challenges and opportunities for the security issues in AI will also be discussed.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; **Adversarial learning**; • **Security and privacy** → **Systems security;**

Additional Key Words and Phrases: Adversarial example attack, artificial intelligence security, poisoning attack, image scaling attack, data collection related attack

This work was supported in part by the National Natural Science Foundation of China under Grant No. 61872130, 62122023, U20A20202, 62002167, and 61874042; the Science and Technology Project of Department of Communications of Hunan Provincial under Grant No. 201928; the Hunan Natural Science Foundation for Distinguished Young Scholars under Grant No. 2020JJ2010, the Hunan Science and Technology Innovation Leading Talents Project under Grant No. 2021RC4019, the Natural Science Foundation of Fujian Province under Grant No. 2021J01544, the Key R & D Projects of Changsha under Grant No. kq1907103; the National Natural Science Foundation of JiangSu No. BK20200461.

Authors' addresses: Y. Hu, W. Kuang, Z. Qin, K. Li, and J. Zhang (corresponding author), Hunan University, Hunan 410082, China; emails: yphu@hnu.edu.cn, wenxinkuang@hnu.edu.cn, zqin@hnu.edu.cn, lkl@hnu.edu.cn, zhangjiliang@hnu.edu.cn; Y. Gao, Nanjing University of Science and Technology, Nanjing, Jiangsu 210094, China; email: yansong.gao@njust.edu.cn; W. Li, New York Institute of Technology, New York, NY 10023; email: wli20@nyit.edu; K. Li, State University of New York, Albany, NY 12246; email: lik@newpaltz.edu.

0360-0300/2021/11-ART20 $15.00
https://doi.org/10.1145/3487890

ACM Reference format:
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, Article 20 (November 2021), 36 pages.
https://doi.org/10.1145/3487890

1 INTRODUCTION

Artificial Intelligence (**AI**) was first introduced at the Dartmouth Conference convened by John McCarthy during the summer of 1956, marking the birth of the AI discipline [97]. However, it was not until 2006, with the introduction of the deep learning concept by Hinton et al. [66], that a new wave of AI applications was ushered in, which was enabled by fast growing computational resources, the emergence of more efficient algorithms, and the explosive growth of data on the Internet.

So far, AI technology has revolutionized many aspects of our daily lives [24, 31, 47, 78, 88, 94, 100, 140, 158, 172] and it empowers us to rethink how we could integrate information, analyze data, and use the resulting insights to improve the overall decision-making process. In order to take the lead in the field of AI, major national strategic plans for AI have been laid out. For instance, the U.S. White House released the "National Strategic Plan for Artificial Intelligence Research and Development" in 2016 [12], while DARPA announced in September 2018 that it would invest nearly $2 billion in the future to develop next-generation AI technologies [147]. In addition, the State Council of China issued the "New Generation Artificial Intelligence Development Plan" [43] in 2017. Nowadays, the level of AI development has become an important reflection of the comprehensive national power of each country.

However, there are bound to be two sides to the development of AI, where its security is becoming a significant concern, especially in security-sensitive infrastructures. According to a leading American technology blog Gizmodo, from 2000 to 2013, 144 people died in surgeries involving robotic assisted surgeons [33]. Statistically, the AI-based recruiting tool used by Amazon from 2014–2017 favored hiring men, raising concerns about the fairness of AI [37]. In March 2018, Uber's automated vehicle crash triggered fears about AI safety [134]. In addition, the image recognition algorithm pulse proposed by Menon et al. [102] has once again sparked great controversy, with someone using the pulse algorithm to restore a blurred image of Obama only to be restored to a white man [154]. According to The Register, a French chatbot based on GPT-3 suggested that model patients commit suicide [119]. In high-risk areas, such as autonomous driving, healthcare, and finance, and so on, a very tiny error or vulnerability could end up costing millions or billions of dollars, and even, sometimes, human lives.

Though the AI systems are generally "smart", they are also "fragile", which means that they can be easily fooled or attacked. Sun et al. [142], Yakura et al. [168], and Zhang et al. [174] discuss AI-related attacks and defenses with graph, audio, and text data, respectively. Chakraborty et al. [23] and Ozdag et al. [113] provide an overview of AI model-related security threats. Unlike other AI security-related reviews that focus on a single type of data or a particular phase of the AI lifecycle, we discuss security threats and countermeasures involving a wide range of typical AI applications, such as image classification, speech recognition, **natural language processing** (**NLP**), and many other scenarios. In addition, we take the AI system lifecycle as a clue to explore and analyze the possible security threats and their defenses at each stage of the AI lifecycle.

The overall framework discussed in this work is shown in Figure 1. It is worth mentioning that MITRE, Microsoft, and 11 other organizations have jointly released the Adversarial **Machine Learning** (**ML**) Threat Matrix [42], an ATT&CK-style framework designed to help security

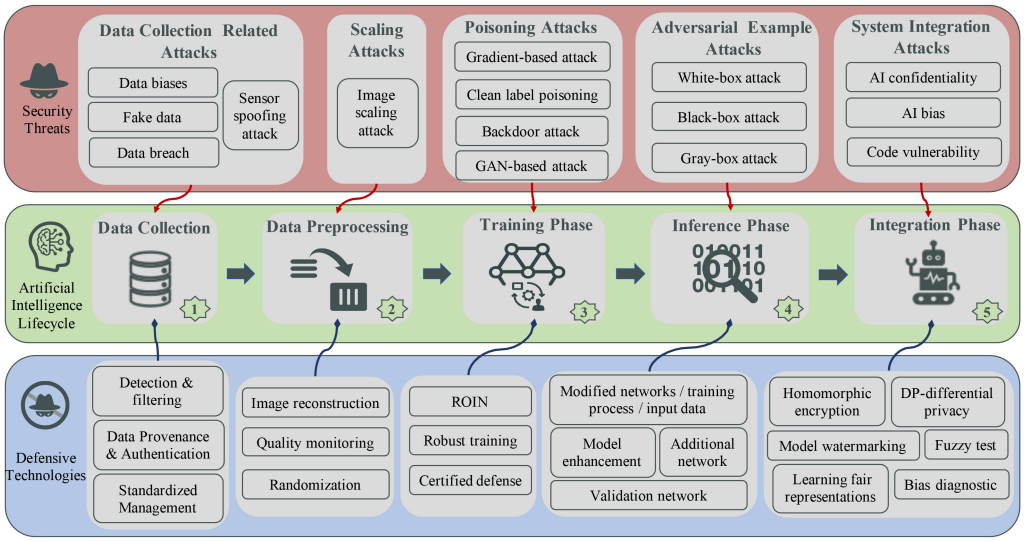

Fig. 1. The overall framework of attack and defense strategies for the AI systems.

analysts quickly locate and remediate attacks on ML systems. The adversarial ML threat matrix is the first attempt toward a knowledge base for ML system attacks and is under initial development. It contains attack techniques specific to ML systems as well as techniques applicable to ML and non-ML systems. Our work can enrich the matrix, especially for ML systems attacks. Since the Threat Matrix is still undergoing refinement, its attack vectors have not encompassed the latest attack techniques, such as sensor spoofing attacks [132] and **image scaling attacks** (**ISAs**) [121, 161]. Furthermore, our framework shows the corresponding countermeasures for different phases of security issues, which can serve as a reference for MITRE to supplement defense techniques in the future. Specifically, the lifecycle of an AI system can be generally divided into five phases: data collection, data pre-processing, model training, model inference, and system integration, each of which is vulnerable to different sets of security threats.

- In the data collection phase, Security risks are closely related to the means of data being collected. There are two main types of data collection methods: software based collection and hardware based collection. One representative attack on the hardware based collection method is the sensor spoofing attack [132], which an attacker performs the sensor attack by accessing or tampering with the data provided by sensors [131, 132]. The software-based data collection method mainly refers to collecting digital data, and its security risks include data biases [37, 111, 154], fake data [30], and data breach [38, 145].
- The data pre-processing phase. Note currently, the scaling attack generally targets the image domain, in which the image data may be tampered with during the pre-processing step, thus becoming a potential attack surface [76, 120, 121, 161]. Specifically, for the insidious ISAs, the attacker tampers with the image and abuses the (visual) cognitive differences between humans and machines to achieve a spoofing and escape attack bypassing even careful manual inspection. Unlike adversarial example attacks, which rely on the model, ISAs only target the data pre-processing step [121, 161]. The attacker utilizes ℓ_p-norm [161] to control the distance between the target image and the attack image to increase the attack success rate. Data randomization [161], quality monitoring [76], image reconstruction [120] are main techniques to defeat the ISA.

— In the model training phase, the causative attacks affect the training data and the training process by injecting poisonous data fed into the model and thus tampering with the trained model. Generally speaking, the causative attack mainly refers to data poisoning attacks [15, 27, 50, 61, 93, 160], which are divided into two groups, namely availability attacks [15, 160] and integrity attacks [61]. For availability attacks, the poisoning points are usually found based on the gradient information [14, 77] of the model, or the poisoning data are automatically generated using an auxiliary network [169]. The availability attack deteriorates the overall performance of the model for any input. In contrast, the integrity attack does not affect the classification of normal inputs but only those attacker-chosen ones. The backdoor attack [62, 151] and clean label poisoning attack [130] are representative integrity attacks. Existing strategies for defending against poisoning attacks include data sanitization [13, 77, 139], robustness training [90], and certified defenses [139].
— In the inference phase, the evasion attacks [2, 4, 18, 22, 55, 136, 148, 159] are normally performed at the model inference phase to degrade or interfere with the model's predictive performance by crafting adversarial examples, which are usually through minor and semantically consistent alterations on the input but without altering the target model [174]. Such attacks have been studied extensively in image classification [18, 55, 159], speech recognition [2, 22, 148], NLP [136], and malware detection [4, 84]. There has been a large number of adversarial example generation strategies developed in recent years, such as classical **Fast Gradient Sign Method (FGSM)** [55], **Jacobian-based Saliency Map Attack (JSMA)** [114], and DeepFool [105] have been developed, which are mainly realized by optimization search or gradient-based information. Correspondingly, countermeasures have been interactively devised, including model-based strategies such as distillation [115], detectors [95], network verification [75], and data-based measures such as adversarial training [55], data randomization [163], and input reconstruction [36, 138].
— In AI system integration phase, security issues become rather complicated. In practical application scenarios, the system integration of AI applications involves not only the security risks of AI technology itself, but also the problems arising from the joint point of the systems, networks, software, and hardware on board. These threats include the confidentiality of AI data and models [48, 175], code vulnerabilities [162], AI bias [64, 154], and so on. The security of AI requires the concerted efforts of researchers in various fields.

In short, the security threats of AI have become an urgent issue in the development and application of AI, especially for security-sensitive scenarios [33, 134]. According to the different phases in the AI system that attacks are targeting, this work elaborates on the corresponding vulnerabilities and their corresponding countermeasures.

2 DATA COLLECTION-RELATED ATTACKS AND DEFENSES

2.1 Overview

Data is the driving force behind the rapid development of AI. It takes many different forms. For example, the types of data include but are not limited to: images and audio captured by hardware devices (e.g., sensors), documents and logs automatically generated by computer systems, and those (e.g., text, image, video, trace) which result from our Internet activities. Furthermore, the security issues involved in data collection are not unique to AI, which essentially exist in any industry requiring data collection. Lin et al. [89] summarized the requirements, objectives, and technologies for data collection related to network security. They argue that data collection needs to meet the following security objectives: confidentiality, integrity, non-repudiation, authentication, privacy protection, and self-protection. However, they recognize that most existing data

Table 1. Methods, Potential Pitfalls, and Defenses of Data Collection

Methods	Security issues	Typical scenarios	Potential defenses
Software-based data collection	Data bias [37, 111, 154]	Social networks, Recommend system	Detection and filtering [64], Standardized Management
	Fake data [30, 155]	Internet of Things, Social networks	Detection and filtering [16]
	Data breach [38, 145]	Scenarios required data collection are covered	Encryption or authentication
Hardware-based data collection	Sensor spoofing attack [79, 131, 132, 135]	Internet of Things	Input filtering [173], Sensor enhancement and Baseband offset [133]

collection technologies meet functional requirements but normally overlooked security goals. Though there is a lack of consensus on the classification of data collection method. It can be generally categorized into software-based data collection and hardware-based data collection [89]. Software-based data collection is in the digital world, while hardware-based data collection is the critical point of converting physical quantities in the physical world into digital forms. Table 1 summarizes the attacks and defenses related to data collection.

2.1.1 Software-Based Data Collection. The daily activities of Internet users generate the majority of data in digital form. Data collectors use software program tools to collect data (e.g., crawlers or "scrapers" of content). Software-based data collection requires packet capture applications, packet capture libraries, operating systems, device drivers, and network cards to work together to complete the data collection process. Theoretically, problems in any part of the process will affect the quality of data collection. We will discuss the security risks caused by software-based data collection methods and their corresponding defenses, using online social networks as an example. Data bias and fake data are representative security risks confronted by social networks' data collection.

2.1.2 Hardware-Based Data Collection. Hardware-related data collection devices include sensors, hardware probes, mobile terminals, data acquisition generation cards, inline taps, network interface cards, mobile terminals, and so on. The potential threat of each type of data collection method varies according to the underlying design principle of the hardware. Sensors are the most widely used data collection tool, and they offer the advantages of efficiency and flexibility. We take the security threats of sensor data collection to illustrate some typical security risks of hardware-based data collection methods.

2.2 Attacks

2.2.1 Data Bias. AI is very sensitive to training data. Data source selection and data preparation may introduce bias [111]. For example, the platform may be driven by business concerns (e.g., specific promotions) or political maneuvers to "nudge" user behavior in social networks. In addition, social platforms discourage third parties from collecting data and impose many restrictions on the **Application Programming Interface** (**API**). As a result, data collectors can only collect limited data or data that are different from what the platform presents to regular users.

AI's incomplete learning bias raises a variety of concerns, such as gender discrimination, racism, and so on. For example, Amazon Human Resources used an AI-enabled recruiting software between 2014 and 2017 [37]. As a result, Amazon hired more male applicants while downgrading the resumes of female applicants. Someone on Twitter used the PULSE algorithm to restore an input blurred image of Obama to a new face with skewed white features [154]. Although not intentional, AI biases undermine the integrity of the AI. We need to improve the data collection criteria and develop tools to diagnose and mitigate bias.

2.2.2 Fake Data. The fake data issue is not a challenge unique to the AI domain. Wanda et al. [155] innovated the pooling function of the convolutional neural network. In addition, they proposed a new dynamic **deep neural network** (**DNN**) model algorithm to detect fake profiles in online social networks. Cobb et al. [30] discussed the security challenges in the data collection process of the data collection application, **Open Data Kit** (**ODK**). They explored the sources of fake data in the IDK data collection process and its defensive measures.

2.2.3 Data Breach. Data breach is a persistent issue. Sweeney et al. [145] first found that only three information fields (place, gender, date of birth) could uniquely identify half of the U.S. population. It is worth noting that data breach is not only a problem specific to the data collection phase but can also occur during the training and inference phases of the model [38].

2.2.4 Sensor Spoofing Attack. Data generated from the physical world need to be digitalized and collected using relevant sensor elements for subsequent model training and inference. Sensors are ubiquitously integrated into, smart wearable devices, automated driving vehicles, and **Light Detection and Ranging** (**LIDAR**), which are the underlying core components responsible for data measurement and collection. Attackers can exploit the physical properties of sensors to construct malicious samples to spoof sensors to interfere with data collection [132]. According to the targeted channel, Shin et al. [131] identified three vectors for sensor spoofing attacks: regular channel, transmission channel, and side channel. Shoukry, Yasser et al. [132] presented a Non-invasive spoofing attack through regular channel. In order to mislead the sensor into producing a malicious speed, the attacker first blocks the magnetic field generated by the rotating gear on the left. The magnetic field generated by the malicious actuator that detects wrong speed is then transmitted to the **Anti-lock Braking Sensor** (**ABS**), which consequentially leads to the sensor spoofing attack. Foo Kune et al. [79] performed a low-power **Electromagnetic Interference** (**EMI**) attack by combining backdoor coupling pair circuit with an analog sensor to perform malicious signal injection. The audio signal is picked up by a microphone. Then the incoming signal is amplified and EMI is injected via an amplifier. After that, they are transmitted to an analog-to-digital converter and subsequently to the microprocessor, which eventually disabled the electronic components. Son et al. [135] attacked the **Unmanned Aerial Vehicle** (**UAV**) with the fact that the output of the gyroscope would fluctuate with the noise at its own resonance frequency. At the resonant frequency of the gyroscope, injecting a specific noise can make the gyroscope resonate, thus deteriorating accuracy and interfering with the operation of the UAV.

2.3 Defenses

Data collection can mitigate security threats by employing data security protection strategies in hardware security, software security, and cyber security. There is a wide variety of data collection protection strategies. Furthermore, the protection strategies vary by scenario. Inspired by data security strategies, we suggest the following three categories as data collection protection measures.

2.3.1 Detection and Filtering. Hinnefeld et al. [64] investigated AI bias and designed a series of strategies (e.g., optimizing preconditioning, rejecting option classification, learning fair representation, and adversarial de-weighting) to detect and mitigate AI bias. To mitigate the threat of data breach, Birnbaum et al. [16] proposed an unsupervised outlier detection technique to detect falsified survey data and illustrated the need to use automated data quality monitoring. In terms of hardware data collection, Zhang et al. [173] conducted software and hardware based defense for different anchor points of attacks. They found that sensor enhancement and baseband offset are useful in defending against sensor spoofing attacks. In the case of microphones, adding a low pass filter while amplifying the microphone amplitude can suppress voice signals above 20 kHz,

which means that human "inaudible" voice commands will be filtered out. Ignjatovic et al. [126] demonstrated that the traditional iterative filtering algorithm aggregating multiple data sources for trust assessment is vulnerable to collusion attacks, so they proposed an iterative filtering technique with better convergence and more robustness to secure sensor networks. We can discard data captured by collection tools that lack trustworthiness and credibility. In addition, attackers can perform a spoofing attack by recording and replaying the command given by users. Although filtering is a convenient and effective defense, we need to be wary of filtering rules that introduce data bias.

2.3.2 Data Provenance and Authentication. Proper sensor trust mechanisms can be adapted to disable the data collected from untrusted devices or unauthorized devices. Firstly, the trustiness of sensor nodes should be checked before aggregating data from them through trustworthiness evaluations [85, 86]. The other commonly used security mechanism is authentication. For instance, a physical challenge-response authentication mechanism, PyCRA, was established by Shoukry et al. [133], in which the sensors use physical probes to continuously and actively sense the surrounding environment. An authentication mechanism is achieved by analyzing the active response to detect manipulated analog signals to defend against malicious sensor attacks.

2.3.3 Standardized Management. Human misuse can also affect the quality of collected data, which requires the management and training of relevant personnel. Therefore, we need to examine security requirements for data collection (confidentiality, integrity, identity verification, etc.) and develop corresponding management procedures to secure data collection [89, 91]. In addition, the establishment of appropriate incentive mechanisms can encourage data providers to share their data more honestly, which is beneficial to the quality of data collection.

3 SCALING ATTACKS AND DEFENSES

3.1 Overview

The size of the image data used to train the models is usually fixed. For example, the image fed into the model is generally sized 224×224 or 32×32, which is smaller than the original—due to the image pre-processing step. In the data pre-processing phase, for example, the images need to be scaled to match the model input size. Image scaling generates a new image with lower/higher resolution in terms of pixels than the original while preserving the original visual features and scaling it proportionally. However, during scaling process, attackers can misuse the scaling algorithm to adjust the pixel-level information to craft a camouflage image, resulting in a dramatic change in visual semantics before and after image scaling. As shown in Figure 2, Xiao et al. [161] have crafted an attack image based on a "sheep" image, which visually disguises the "wolf" as a "sheep." Once the image is down-sampled or resized, the real "wolf" is revealed. Moreover, Xiao et al. [161] verified the effectiveness of the attack on multiple cloud-based image servers like Microsoft Azure, Aliyun, Tencent, and Baidu image classification service. For instance, Baidu cloud server identified the image as a "wolf" with high confidence. Notably, the ISA is powerful to be agnostic to different models as long as they employ the same rescaling function to fit the same model input size. Table 2 presents an overview of ISAs and defenses in the data preprocessing phase.

3.2 Attacks

Xiao et al. [161] first revealed ISA by exploiting the inverse of the interpolation algorithm. As shown in Figure 3, a perturbation matrix Δ_1 is first added to the original image "srcimg" e.g., digit number 8 to produce the attack image "attackimg" that embeds the targeted image e.g., digit number of 6. while Δ_2 is the difference between the target image "targetimg" and the output image

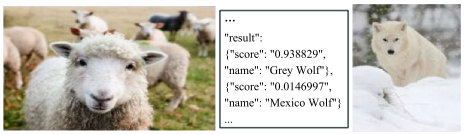

Fig. 2. Example of a scaling attack [161].

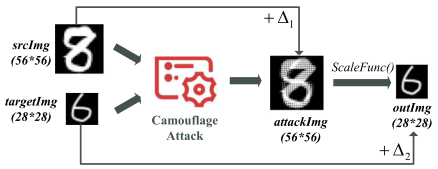

Fig. 3. Automatic attack image crafting [161].

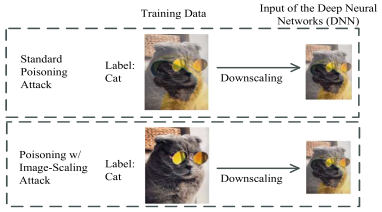

Fig. 4. A clean-label poisoning attack [121].

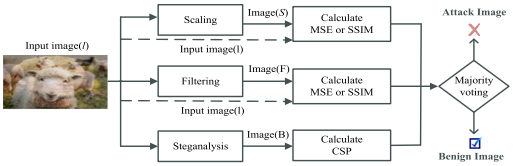

Fig. 5. Overview of Decamouflage [76].

"outimg." Finally, the optimal attack image is generated based on the interpolation algorithm in the constraints of $\Delta_1$and Δ_2. Once the image-scaling operation is routinely performed on attack image, the model sees the targeted image e.g., digit number of 6, and thus identifies it as the attacker's target of 6, which is a source-to-target attack [161]. Such an attack is still effective even when the deployed system is a black-box to the attacker, as it is relatively easy to infer required parameters such as the input image size or/and the underlying rescaling function used by that model through e.g., exhaustive trials. This is due to the fact that the commonly used type of input image sizes or/and rescaling functions are limited.

The root cause of the ISA is from the interaction of downsampling and convolution, which is theoretically analyzed by Quiring et al. [120] from a signal processing perspective. They conducted experiments on three ML imaging libraries (OpenCV, TensorFlow, and Pillow) to confirm the existence of such interaction. By making the attack image consistent with the scaled image in the color histogram, Quiring et al. [121] introduced a new Adaptive ISA, which reduced the success rate of ISA detection by inspecting the color histogram.

In addition, the authors combined the ISA with the poisoning attack, successfully hiding the triggers of the backdoor attack [121]. As illustrated in Figure 4, the ISA technique is used to hide the triggers before data pre-processing, ensuring that the content and labels of the attack images with triggers are visually consistent, thus bypassing the manual inspection of poisoned images. Once the downscaling operation, as a standard step in most cases, is executed, the trigger is immediately exposed. Such poisoned samples via image-scaling ensure the concealment of the trigger and achieve the effect of the backdoor attack.

3.3 Defenses

Quiring et al. [120] developed an image reconstruction method to defend against ISAs. Selective median filters and random filters are employed in their work to identify pixel points that have been altered during scaling. Then the remaining pixels in the image are used to reconstruct modified contents e.g., through median. The defense method proposed by Quiring et al. [120] avoids to modify the original neural network, but simply combines with existing image libraries to defend against scaling attacks.

Image reconstruction prevents ISAs by the relationship between downsampling frequency and image scaling, but degrades the quality of the input image [120]. Therefore, as illustrated in

Table 2. Image Scaling Attacks and Defenses

Attacks	Attack strategy	Potential defense
Xiao [161]	Reducing the ℓ_p-norm distance between the target and attack image	Randomization, Quality Monitoring
Quiring [120]	Reducing the color histogram gap between the target and attack image	Image reconstruction [120], Attack detection [76]

Figure 5, Kim et al. [76] integrated rescaling, filtering and steganalysis into a scaling attack detection framework—decamouflage. Specifically, (i) Scaling detection method firstly performs downscaling and then upscaling operations on the input image to construct a "copy" image, then compares the similarity of the image on the color histogram before and after between the input image and its "copy": The attacker injected pixels in the input image are expected to be removed from the "copy" attributing to the upscaling. (ii) Filtering detection filters the image using filters. (iii) The samples suspected to attack images are transformed into two-dimensional space by **Discrete Fourier Transform** (**DFT**) and perturbed pixels embedded by ISA are detected using steganalysis. Subsequently, the **Mean Squared Errors** (**MSE**), **Structural Similarity Index** (**SSIM**), and **Centered Spectrum Points** (**CSP**) metrics are used to quantify the similarity before and after, and the derive detection boundaries for each detection method independently. Finally, an ensembled technique is performed to identify whether the incoming image is an attack image or not.

In general, Quiring et al. [120] eliminated the attack effect but does not specifically detect whether the input image is an attack image. Whereas, Kim et al. [76] detected the presence of malicious attacks and rejects the attack images. Detection is preferred in case that the any attack is required to be traced. In addition, it is possible to adapt [120] to further eliminate the attack effect by reconstructing the attack image to get the correct prediction once the input is detected as adversary, which can alleviate quality degradation caused by the image reconstruction in [120].

4 DATA POISONING ATTACKS AND DEFENSES

4.1 Overview

AI systems are trained based on large curated data. However, the data quality directly affects the performance of the model trained. In this context, an attacker can poison the training set to manipulate the inference behavior of the model. From the perspective of the model and the attack target, poisoning attacks can be divided into two categories: availability attacks [15, 160] and integrity attacks [61].

- Availability attacks are known as denial-of-service attacks, where the attack goal is to maximize the overall loss of the model and cause a degradation in model performance as well as misclassification. For example, social media chatbots have a rich corpus and which is expanded by interactions with humans. When an attacker affects the chatbot with some statements with no contextual relevance, the chatbot will not conduct a normal logical chat.
- Integrity attacks are where the attacker accomplishes targeted damage by carefully designing poisoned data without affecting the model's classification of clean samples [27, 61, 93]. The most representative integrity attack is the backdoor attack. Backdoor attacks only misclassify inputs containing specific (explicit or even inexplicit) triggers and the backdoor can still be retained in a downstream transfer learning task. As an example of backdoor attack, in malware detection, an attacker marks a file containing a specific string as benign data and puts it into the training of the detector. After the model is trained and deployed, the attacker simply adds the particular string to the malware to evade detection, as any malware with the specific string functioning as a trigger will be associated with the benign class.

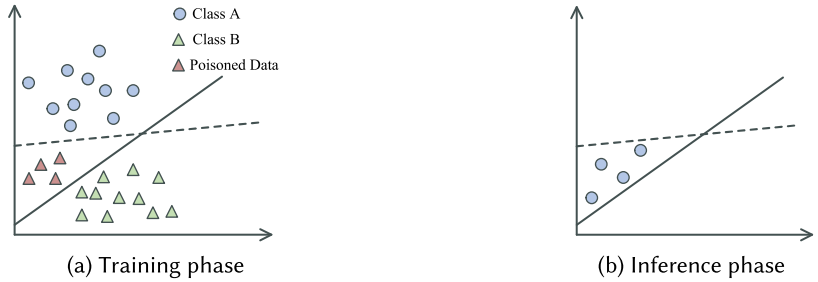

Fig. 6. Changes in classification models before and after poisoning attacks.

Based on the differences in attack behavior and classification results, poisoning attacks are divided into Error-Specific attacks and Error-Generic attacks. Suppose there is a clean sample **C** with the true label y_{true}. The attacker constructs a poisoned sample set C' and adds it to the training set of model M, causing model M to misclassify C, i.e., $M(C) \neq y_{\text{true}}$. If $M(C)$ is a specific class targeted by the attacker, it is an Error-Specific poisoning attack. Whereas, if $M(C)$ is any class other than y_{true}, it is an Error-Generic poisoning attack. As shown in Figure 6(a), the solid line indicates the binary classifier under normal conditions. Suppose a small amount of poison data is added to training sets. In that case, the decision boundary will be shifted, resulting in a classification effect as separated by the dotted line. Therefore, instances within the closed region formed by the intersection of the normal model and the poisoned model will be misclassified during the inference phase. As shown in Figure 6(b), Class A instances will be misclassified as Class B.

4.2 Attacks

Next, we will detail various poisoning attack methods, which we also have summarized in Table 3.

4.2.1 Availability Attacks. Availability attacks are known as denial-of-service attacks. Representative availability attacks include gradient-based attacks and **Generative Adversarial Network** (**GAN**) based. Availability attacks with poisoning can be formally represented as a bi-level optimization problem [14, 15, 99]. The inner optimization is a model training problem on a poisoned training set. The outer optimization is to maximize the attacker's objective **A**, which is usually the classification loss function **L** of the clean dataset on the poisoned model obtained from the inner optimization. The formal representation is as follows:

$$D_c^* \in \underset{D_c' \in \Phi(D_c)}{\operatorname{argmax}} \quad A\left(D_c', \theta\right) = L\left(\widehat{D}_{val}, \widehat{w}\right), \qquad \text{s.t.} \quad \widehat{\boldsymbol{w}} \in \underset{\boldsymbol{w}}{\operatorname{argmin}} l\left(\widehat{D}_{tr} \cup D_c', w\right). \tag{1}$$

Attackers have only access to a proxy dataset $\widehat{D}$ of the data source. $\widehat{D}$ is divided into two disjoint subsets $\widehat{D}_{tr}$ and $\widehat{D}_{val}$. The $\widehat{D}_{tr}$ and the poisoned sample set D_c' are used to train the model to obtain the poisoning model parameter $\widehat{w}$. $\widehat{D}_{val}$ is used to test the classification effect on clean dataset in the surrogate model by a simple loss function $L(\widehat{D}_{val}, \widehat{w})$. In other words, the effect of poisoned samples on clean data is determined by the parameter $\widehat{w}$.

Gradient-based Attacks. The major challenge of gradient-based poisoning attack is the calculation of the gradient $\nabla_{x_c} A$ of the attack target with respect to the poisoning point. In general, both the gradient-based ascent and reverse gradient optimization obtain the optimized poisoning point x by calculating the gradient of the attack target with respect to each poisoning point.

— Gradient-based ascent. The gradient ascent poisoning attack technology is optimized by adopting a gradient ascent approach [14, 77]. Assuming that the attack function $A\left(D_c', \theta\right)$

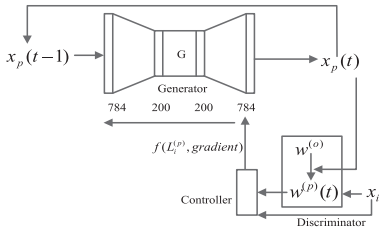

Fig. 7. An overview of the GAN-based poisoning method [169].

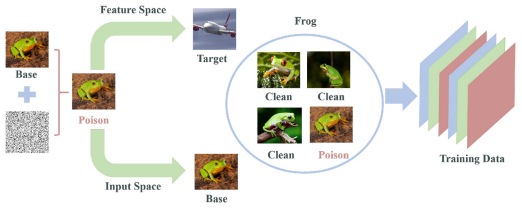

Fig. 8. An example of clean label poisoning.

is differentiable for parameters w and input x, the required gradient is calculated using the chain rule as follows:

$$\nabla_x A = \nabla_x L + \frac{\partial w^T}{\partial x} \nabla_w L, \tag{2}$$

where $\frac{\partial w}{\partial x}$ denotes the hidden dependence of the classifier parameters on the poisoned data. Mei et al. [99] proposed an implicit equation that uses the **Karush–Kuhn–Tucker** (**KKT**) condition instead of the inner optimization problem to derive the gradient. By differentiating at the poisoned point, the gradient can be solved. Then, a two-layer optimization problem is transformed into a single-layer constrained optimization problem. Although the optimization is simplified, the complexity of gradient computation makes it only applicable to a limited number of learning algorithms.

— Reverse gradient optimization. Reverse gradient optimization by Muñoz-González et al. [109] is the first poisoning attack against a deep learning framework. They update the parameters by reversing the learning process. The inner optimization problem is replaced by learning iterations. The required gradient in the external optimization problem is obtained through the incomplete parameters in the inner optimization problem. They Assume that one poisoning point x_c is optimized at a time. In the inner optimization problem, a total of T iterations are performed. Thus, the parameter W_T is obtained. The chain rule is used to calculate the gradient for updating the poisoning points.

GAN-based Attacks. Yang et al. [169] designed a generator inspired by GAN to accelerate the production of poisoned samples. The generator firstly randomly selects a sample x_i from the clean training set D_t for yielding a poisoned sample. Then, the discriminator uses the poisoned data generated by the generator to calculate the loss of the clean data. Subsequently, the generator updates the poisoned data using the new weighting function of gradient and loss provided by the discriminator. The process is iterated continuously until the termination condition is reached. Figure 7 illustrate an overview of the GAN-based poisoning method. The target model acts as a discriminator while the generator is an additional model designed to generate the poisoning data x_p. The poisoning data obtained from the $(t-1)_{\text{th}}$ update is input to the generator to obtain the poisoning data $x_p(t)$ updated in the t iteration. Then, $x_p(t)$ is injected into the discriminator and the weighted gradient $f(L_i^{(p)}, \text{ gradient})$ is calculated to update the target model. Only one update of the target model is required for each iteration, which can greatly reduce the time of poisoning data generation.

4.2.2 Integrity Attack. Integrity attacks can accomplish targeted damage without affecting the model's classification of normal samples. Backdoor attacks are the most representative integrity attacks.

Backdoor Attacks. Backdoor attacks [62] do not affect the results of clean data being classified in the backdoor model, but will produce deviations from the expected results for inputs containing specific triggers secretly controlled by an attacker. Backdoor attacks is a typical integrity poisoning attack by adding triggers to clean samples for creating poisonous samples whose labels are usually modified into the targeted label. Significantly, the trigger, such as its position, shape, or color, can be under the arbitrary control of the attacker. For an input x, its poisoned counterpart $A(x, m, \Delta)$ is obtained by stamping a trigger. Take image domain as example, m represents the trigger position. Δ denotes the trigger color, pattern, and other information. The final trigger optimization problem minimizes the dissimilarity of the latent representations of two models in the feature space, which can be formally described as

$$\Delta^f = \underset{\Delta}{\operatorname{argmin}} \sum_{x \in X} \sum_{x_t \in X_t} D\left(F_\theta\left(A(x, m, \Delta), F_\theta\left(x_t\right)\right)\right), \tag{3}$$

where t denotes the target attack class, x_t denotes input of the target attack class. $F_\theta(x)$ denotes the output of x under parameter θ or the intermediate output of a certain layer in neural network. Since the trained triggers will activate some specific neurons in the model, $D(\cdot)$ is utilized to measure the difference in neuronal activation states for clean inputs and inputs with added triggers. The commonly used $D(\cdot)$ is the MSE. Gu et al. [61] proposed a backdoor attack against neural networks, where each neuron can be regard as an inner feature. The layer between the layer where the selected neuron is located and the output layer is retrained so that the trigger establishes a strong connection with the target class of neurons in the output layer.

Turner et al. [151] considered a backdoor attack in which the injected poisoned samples are visually consistent with the label. To maintain the consistency of the label, they modify the pixel value of the original backdoor pattern by the backdoor trigger magnitude so that the backdoor trigger pattern is visually inconspicuous. Experiments show that this approach can generate an unobtrusive trigger and be learned by the model to achieve a successful backdoor attack. Barni et al. [9] added backdoor by corrupting the target class sample data. Once a backdoor signal is encountered, the network identifies the sample as the target class. The method allows selecting appropriate perturbations according to different classification tasks and target classes. For example, for the MINIST digital classification task, they define the backdoor additive perturbation based on the slope signal as $v(i, j) = \frac{j\Delta}{m}, 1 \le j \le m, 1 \le i \le \ell$. m and l are the number of columns and rows of the image, respectively. The backdoors formed by this method are more stealthy and have higher attack success rate. However, they only corrupt the target class samples and need to increase the data poisoning ratio to achieve high attack success rate.

Clean Label Poisoning. Shafahi et al. [130] proposed clean label poisoning attack that retains the consistency between the label and visualize content of the image. In brief, they make the model decision boundary change by adding poisoned data (labeled as base class) to the training set, causing clean target instances around the poisoned data to be misclassified as base class. The attack is depicted in Figure 8. The target class and the base class are firstly determined. Then a target instance t and a base instance b are selected from the target and base class, respectively. A poisoned sample x is constructed under the ℓ_2-norm constraint in a way that x is visually similar to the base class but close to the target class in the feature space representation. Poisoning data generation by feature collision is formulated as

$$p = \underset{x}{\operatorname{argmin}} \|f(x) - f(t)\|_2^2 + \beta \|x - b\|_2^2, \tag{4}$$

where $f(x)$ is the representation of x in the penultimate layer of the model, called the feature space representation of x—feature space representation. $\|f(x) - f(t)\|_2^2$ is the ℓ_2-norm measuring

Table 3. Data Poisoning Attack Methods

Category	Attack Strategy	Advantages	Disadvantage
Availability attack	Gradient-based [14, 77, 109]	Model robustness enhancement	High computational complexity
	GAN-based [169]	Low time overhead, low complexity, model irrelevant	Poor generalization
Integrity attack	Backdoor [9, 62, 151]	Does not affect normal sample classification, model irrelevant, highly concealed, high generalizability	Scenarios limited to image recognition, need to retrain or introduce additional models
	Clean label poisoning [130]	Easy implementation	High computational complexity

Table 4. Data Poisoning Defense Methods

Defense strategy	Advantages	Disadvantages
Data sanitization [77, 110]	High generalizability, easy implementation, high detection success rate in certain scenarios	Easy overfitting for small samples, high computational overhead
Robust training [71, 90]	Model robustness enhancement, low complexity	High computational overhead
Certified defenses [139]	High interpretability	High complexity

the feature space similarity to the target instance, and β regulates the visual similarity of the poisoned sample x to the base class from the raw input space. The optimization problem is solved by a forward–backward splitting iterative process. To be precise, the first step (forward) minimizes the ℓ_2 distance between the target instance and the poisoned instance in the feature space. The second step (reverse) is to minimizes the distance between the poisoned data and the base instance in the input space. The optimization following Equation (4) can provide a set of poisoned images that look like the base class but are consistent with the target class in the deep feature space so that the base class label needs no change.

4.3 Defenses

Data poisoning attacks inject poisoned data into the training set to disrupt the functionality of learning algorithms. Poisoned data have different characteristics from clean data, which means that poisoned data can be treated as anomalies so that anomaly detection can be used as a defense. Data sanitization [13, 34, 77, 110, 139] usually applying anomaly detection or model robustness training [17, 28, 71, 90, 127] can be adopted to defend against the data poisoning attacks. We summarize these defenses in Table 4.

4.3.1 Data Sanitization. Nelson et al. [110] presented **Reject On Negative Impact** (**ROIN**) against data poisoning attacks on spam filters. If the data have a significant negative impact on the classifier, it is treated as poisoned data and removed from the training set. Although ROIN has shown excellent performance in defending against data poisoning attacks in certain scenarios, such as identifying attack e-mails with a 100% success rate, it is too expensive to test every data sample in the training set. Moreover, overfitting is prone to occur when the dataset is smaller than the number of features. Koh and Liang [77] applied the influence functions in robust statistics to calculate the effect of data points on the prediction of the classifier. The method proposed by Koh and Liang was able to determine the influence of each data item without retraining the model—ROIN [110] requires retraining the model—using only the gradient and the Hession matrix, which ensures that the data points impairing the performance can be quickly identified.

4.3.2 Robust Training. Robust training generally strongly relies on some feature assumptions. Liu et al. [90] relaxed the assumptions and achieved strong defensive performance by improving robust low-rank matrix approximation and robust principal component regression. Jagielski et al. [71] designed an adversarial defense technique called "TRIM" by using a pruning loss function

to compute different subsets of residuals in each iteration for robust training of linear regression models. In general, the **Support Vector Machines** (**SVMs**) are not robust to outliers. Xu et al. [164] improved the correntropy induced loss function and constructed the rescaled hinge loss function to extend the robustness of SVM.

4.3.3 Certified Defenses. Steinhardt proposed a certified defense against poisoning attacks [139]. A framework was designed for defenders who adopt anomaly exclusion and empirical risk minimization, aiming to study the entire attack space for a given defense. Assume that D_c and D_p denote the clean and poisoned datasets, respectively, and θ denote the parameters of the classifier. The corresponding defenses are designed for scenarios where the feasible set is dependent on the poisoned data D_p or not. In the case of independence on D_p as an example, they proposed a fixed defense approach. In the iterative solution process, the current worst attack point $(x^{(t)}, y^{(t)}) = \text{argmax}_{(x,y)\in\mathcal{F}}\, \ell(\theta^{(t-1)}; x, y)$ is found first each time. Then the model is updated in the direction of that attack point to get $\theta^{(t)}$. Eventually, the worst poisoning attack dataset D_p can be found. The worst-case validation error upper bound M is found based on D_p, which is approximated as the training error on the entire dataset (clean and poisoned data). Anomalies on a clean dataset do not unduly affect the model.

4.3.4 Other Defenses. To mitigate the impact of backdoor attacks, a particular class of data poisoning attacks, Wang et al. [156] proposed Neural Cleanse by exploiting the principle that the trigger is eventually the (abnormal) smallest perturbation required to tamper all images to the targeted class. Therefore, Neural Cleanse identifies such smallest perturbation to reverse engineering the trigger, which can be consequentially used to unlearn the backdoor for removal. Liu et al. [92] pruned redundant neurons in neural networks that were not sensitive to classification while using clean data to fine-tune the models, thus allowing them to classify properly. However, their approach assumes that all models are potentially implanted with backdoors, and blindly performing pruning fine-tuning on models tends to degrade the accuracy of normal models performing normal tasks. Chen et al. [25] detected poisoned data by activation clustering techniques based on the difference in the activation status of neurons in the neural network between poisoned and original data. Gao et al. [51] proposed STRIP to detect trigger inputs during run-time without any ML technique. The principle is, for the input-agnostic backdoor attack, that the trigger input will always be classified into the targeted label regardless of the input content. This is because the trigger fully hijacks the model. Therefore, when the strong perturbation is added to the trigger input, the prediction is less influenced: insensitive to perturbation. However, the normal input should be sensitive to strong perturbations. So that examination of the randomness of predictions of a set of perburbed replica of inputs can distinguish trigger input and normal input: trigger input exhibiting low randomness, while normal input exhibiting high randomness.

5 ADVERSARIAL EXAMPLE ATTACKS AND DEFENSES

5.1 Overview

Adversarial example attack performed in the inference phase is the most studied security threat of AI system. Suppose there is a clean sample x with $f(x) = y_{\text{true}}$ predicted by a trained model M. The attacker adds an imperceptible perturbation δ to x to create an adversarial example $x' = x + \delta$, which is sufficient to mislead the model to produce a wrong output $f(x) \neq y_{\text{true}}$, also known as untargeted attacks. In terms of targeted attack, it is to mislead the model to yield an attacker targeted class $f(x) = y_{\text{target}} \neq y_{\text{true}}$. Due to the lack of interpretability and complexity of DNNs, there is still no uniformity regarding the causes of AI vulnerability to adversarial attacks. Some researchers argue that the leading cause of adversarial attacks is the highly non-linear nature of

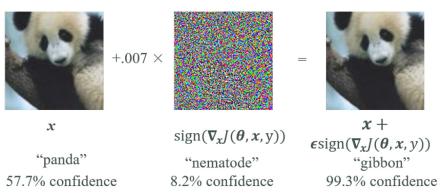

Fig. 9. Overview of the FGSM adversarial example generation [55].

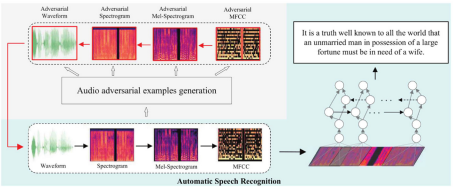

Fig. 10. Overview of the process of adversarial attacks and ASR [67].

the model [116], while Goodfellow et al. [55] believed that it is caused by the linear behavior of the model in a high-dimensional space. A relatively recent study by Ilyas et al. [70] indicates that the adversarial example is closely related to the non-robust features, which are highly predictive, yet brittle and (thus) incomprehensible to humans. The problem of adversarial example attack has been demonstrated in various AI-related fields below.

5.1.1 Image Classification. In the image domain, the aim is to add perturbations slightly to the original pixels that are unrecognized with naked eyes while misleading models [18, 35, 45, 55, 80]. The concept of adversarial examples was first introduced by Szegedy et al. [146]. They found that the semantic information (a certain feature) contained in the high level of neural networks is distributed across the spatial structure of the network rather than individual neurons, and the mapping between the input and output of the neural network is mostly discontinuous. Therefore, adding disturbances of the same magnitude to the same input can render different neural networks to make similar misclassifications. Specifically, neural networks have certain blind spots, and regular perturbations can be injected into the input image to fool the network. **Fast Gradient Sign Method (FGSM)** by Goodfellow et al. [55] is an early representative adversarial example generation algorithm in the image field. As an example in Figure 9, for the original input x, the established model identifies it as panda y. After adding carefully crafted human imperceptible noise, the network outputs a gibbon y with a 99.3% probability of not matching the original class, although it is a panda image discernible to the naked eye. Since the pixel values representing the image are approximately continuous, the similarity between the fake image and the legitimate image can be discerned artificially and intuitively. In other areas like speech recognition, NLP, and malware detection, data and classifier structures are more complex, which requires more caution when mounting adversarial example attacks.

5.1.2 Speech Recognition. **Automatic Speech Recognition** (**ASR**), which requires filtering and digitization operations on the raw audio before acoustic features can be extracted, is a technology that enables intelligent devices to recognize and understand human speech or/and convert it into text [112]. Besides MFCC, the DFT and the **Fast Fourier Transform** (**FFT**) can also extract speech features. The framework of ASR is shown in Figure 10. Firstly, by dividing the speech signal into several blocks, which could be overlapped. Then, the spectrum map is obtained by calculating the amplitude of the spectrum for each block through FFT, and the Mel-Spectrum is obtained by performing Mel-Filters on the spectrum. Next, the MFCC is derived by cepstrum analysis [108]. Finally, the obtained MFCC is input to the neural network for recognition/classification. The Mel-Filters serves to deform the frequency to follow the spatial distribution of human ear hair cells. The end-to-end ASR systems [3, 32, 123] usually utilize the **Connectionist Temporal Classification** (**CTC**) loss function [57] to derive characters but not phoneme sequences directly. The gap between human and machine speech recognition generates unmonitored channels through which adversarial examples can be implanted with commands. Compared with the image classification

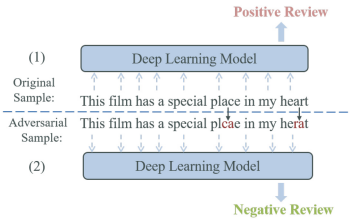

Fig. 11. Discrete perturbation in text processing schematic [49].

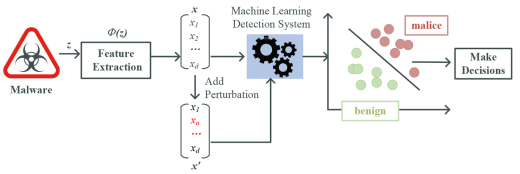

Fig. 12. Malware detection confrontation example.

domain, the adversarial examples of ASR systems are more difficult to craft since some common audio processing operations are highly prone to introduce additional noise.

5.1.3 Natural Language Processing. NLP is to recognize human language by computers. Applications of NLP range from input recognition, tag classification to analytical understanding and processing of words and sentences [54] as well as chapters [73, 96] of documents. Adversarial example attacks also exist in NLP, though there are fewer studies than image and audio domains. Pixel-level adversarial attack methods cannot be directly mounted for generating in NLP due to the differences between image pixels and text data. Firstly, image data (e.g., pixel values) are continuous in the numerical profile, but text data label types are discrete. In general, text data needs to be vectorized before feeding into a DNN. Word embeddings are usually used as input to DNNs. However, the networks can fail to match the word embedding space [54]. Secondly, the perturbation to the image is a modification at the pixel value level difficult to perceive. While for subtle textual perturbation, it is extremely vulnerable to be detected. As exemplified in Figure 11, slights modification of the characters of words in the original input sample (1) can result in invalid words and thus affect the overall semantics of the sentence to a large extent. Therefore, for text-type adversarial example attacks, Most studies focus on tasks of reading comprehension than short texts.

5.1.4 Malware Detection. Malware detection is the process of classifying software features extracted by static or dynamic analysis using AI technology. Static analysis extracts and analyzes the features of malware samples without execution. Dynamic analysis, on the other hand, requires executing them and analyzing their corresponding features. Commonly used tools for dynamic analysis include sandboxes, simulators, and so on. [170]. Features commonly used in malware detection are byte sequences, opcodes, APIs and system calls, network activity, file systems, PE files, and so on [152]. As in Figure 12, firstly, feature sequences that represent malware are filtered by feature extraction. Then, a malware classifier is trained over the dataset consisting of the feature sequences. The adversarial example attacks aim at adding some functionally independent features to the feature vector to generate malware adversarial examples.

5.2 Attacks

The adversary attack capability is determined by how much information the attacker has about the model, including training data, feature sets, learning algorithms, and so on. Based on available knowledge by the attacker, attacks may be divided into three main groups: white-box, black-box, and grey-box.

— White-box attack: The attacker is with full knowledge about the target model, including the type of neural network model, the parameters and training algorithm, and so on. The adversary applies known knowledge to identify vulnerable feature spaces to facilitate the generation of adversarial examples. Since white-box attacks require computing the gradient

Table 5. Adversarial Examples Generation Methods

Method	Scenarios	Strengths and weaknesses	Attack specificity	Category
L-BFGS [146]	Image classification	Optimization-based search, high computational complexity	targeted/ untargeted	white-box
FGSM [55]	Image classification Speech recognition Malware detection	Gradient-based, single iteration, simple computation, large perturbation	targeted/ untargeted	white-box
I-FGSM/ BIM [80]	Image classification	Iterative FGSM, more efficient	targeted/ untargeted	white-box
ILCM [80]	Image classification	Make the probability of the minimum class increase, iterative solution, essentially similar to BIM	targeted	white-box
JSMA [114]	Image classification Natural language processing Malware detection	ℓ_0 attack with large perturbation, suitable for black-box migration attack, requires microscopic target model, difficult to be realistic	targeted untargeted	white-box
Deep fool [105]	Image classification	Based on decision surface, small perturbation, low complexity; low success rate of black box attack	targeted/ untargeted	white-box
C&W [21]	Image classification Natural language processing Malware detection	Gradient optimization solving, with high complexity	targeted/ untargeted	white-box
UAP [104]	Image classification	No need to solve optimization problems, gradient calculations	untargeted	white-box
MalGAN [69]	Malware detection	Gradient-independent; high attack success rate, high overhead	–	black-box
EvadeML [167]	Malware detection	Gradient-independent, high attack success rate, not easily scalable	–	black-box
ATNs [124]	Image classification	Can attack one or more networks; higher training costs	targeted/ untargeted	white-box/ black-box
ZOO [26]	Image classification	Approximate gradient estimation; high success rate, more expensive to query and estimate gradients	targeted/ untargeted	black-box
Houdini [29]	Image classification	Spoofing gradient	targeted/ untargeted	black-box
One pixel [141]	Image classification	Single-pixel perturbation, no gradient information, heuristic solution, low efficiency	targeted/ untargeted	black-box

with respect to the input, while the gradient is discrete in the textual case, white-box attack methods based on gradient optimization are challenging to be applied to NLP.

— Black-box attacks: Contrary to white-box attacks, the adversary does not know any knowledge about the model but is allowed to analyze the vulnerability/weakness of the model by querying the AI system with carefully devised inputs and observing the outputs. Black-box attacks are more practical but more challenging to design.

— Grey-box attacks: The scenario of grey-box attack was introduced by Meng et al. [101]. It is also known as semi-white box attacks [165]. The attacker needs to obtain partial knowledge of the model (except for the model parameters) to complete the attack on the target model [142]. Grey-box attacks are not common in practice [174].

Though this work does cover adversarial example attacks in all fields for the sake of illustrating the current typical adversarial example construction methods, the image classification that has been widely used in practice is the main focus, as shown in Table 5. Beyond that, some attacks in other areas are briefly discussed in Section 5.2.3.

5.2.1 White-Box Attacks.

L-BFGS. Szegedy et al. [146] first demonstrated that a small amount of perturbation imperceptible to humans to an image could mislead neural networks. However, the complexity of solving

the optimization problem for the minimum perturbation was too high, so they opted Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) for the approximate solution by finding the minimum. Concretely, the problem of constructing an image x' similar to x: *minimize* $\|\nabla\|_2^2$ by transforming it into a convex optimization problem with the ℓ_2-norm constraint, is formally expressed as $\min c\,\|x - x'\|_2^2 + J_\theta\,(x', l)$ s.t. $x' \in [0,1]^n$. The hyperparameter $c > 0$. The adversary tries to constrain the similarity between the adversarial example and the normal sample by the regularization function based on ℓ_p-norm while using the loss function $J_\theta\,(x', t)$ to make x' be misclassified as the target class t. The adversarial example after perturbations are still in the normal image fetch range $x' \in [0,1]^n$. Meanwhile, they showed that introducing adversarial examples into the training process can improve the model generalization.

FGSM. Goodfellow et al. [55] devised FGSM to calculate adversarial perturbations. Compared with L-BFGS, the calculation of FGSM only needs to be performed in the backpropagation stage, so its generation of adversarial examples is faster and more suitable for scenarios where a large number of adversarial examples need to be generated. Suppose x and y are the original image and the corresponding label, respectively. $J_\theta(x, y)$ is the loss function. Based on the difference in attack behavior, FGSM is further classified into targeted and non-targeted attacks. The FGSM attack by Goodfellow et al. [55] can be represented as Equation (5). Kurakin et al. [81] extended it to targeted attack as Equation (6).

$$x' = x + \varepsilon \times \text{sign}\left(\nabla_x J_\theta(x, y)\right), \text{ non-target}, \tag{5}$$

$$x' = x - \varepsilon \times \text{sign}\left(\nabla_x J_\theta\,(x, y')\right), \text{ target on } y'. \tag{6}$$

Take the targeted attack, for example. FGSM first identifies the class with the lowest probability as the target. Subsequently, the original image is subtracted from a set perturbation, thereby generating an adversarial example sufficient for misleading the model to output the target class. The FGSM has been recently improved by a number of studies, such as R+FGSM [149] and MI-FGSM [39]. Notably, FGSM has not only made remarkable achievements in the field of image classification but also in the field of speech recognition. Previous studies have explored how acoustic features can be extracted to generate audio adversarial examples, Gong et al. [53] proposed the first end-to-end audio adversarial example generation method based on gradient sign.

BIM & ILCM. FGSM generates adversarial examples by adding perturbations to each pixel point in the gradient direction in a non-iterative manner. In contrast, the **Basic Iterative Method** (**BIM**) Iterative FGSM (I-FGSM) by Kurakin et al. [80], uses an iterative approach to find the perturbation for each pixel point. BIM is represented formally as follows:

$$x'_0 = x, \quad x'_{i+1} = Clip_{x,\epsilon}\left\{x'_i + \alpha \times \text{sign}\left(\nabla_x J\left(x'_i, y_{true}\right)\right)\right\}. \tag{7}$$

Each time, the individual pixels grow (or decrease) in steps of α based on the previously generated adversarial example x'. A cropping operation $Clip_{X,\epsilon}\{x'\}$ is then performed to constrain the individual pixels of the new example from deviating too much from the original image x. Ideally, the adversarial example can be found with a variation of less than ε for each pixel. In the worst case, the same effect as FGSM can be achieved. In addition, Kurakin et al. [80] designed the Class Iterative Methods Method (ILCM) by replacing the class y_{true} in the adversarial perturbation with the lowest probability class y_{LL}.

JSMA. Papernot et al. [114] scrutinized the relationship between the model input features and output results and introduce the JSMA adversarial example generation algorithm. It consists of three main processes as follows:

(1) Compute the forward derivative $\nabla F(\boldsymbol{X}) = \frac{\partial F(\boldsymbol{X})}{\partial \boldsymbol{X}} = [\frac{\partial F_j(\boldsymbol{X})}{\partial X_i}]_{i\in 1,M,j\in 1,N}$.
(2) Calculate the adversarial saliency map S using the results of (1). A higher value of the saliency map indicates that the corresponding input features significantly influence the classification result.
(3) Select significant pixels for disturbance according to the saliency map in (2). The pixel values of the selected pixel points are increased or decreased and then iterated until the model output them as the target class.

JSMA is stealthy and only needs to perturb some important pixel points rather than the whole image. In addition, JSMA has been used in NLP and malware detection. Grosse et al. [59] employed the Jacobian matrix to generate adversarial examples of Android malware to escape the detection of a DNN-based malware detector.

C&W. Carlini and Wagner [21] suggested an optimization-based algorithm C&W to attack defensive distillation networks by setting a loss function that measures the difference between the inputs and outputs. The loss function contains an adjustable hyperparameter and a parameter controlling the confidence of adversarial examples, which can be formally expressed as $f(x') = \max(\max\{Z(x')_i : i \neq t\} - Z(x')_t, -\kappa)$. Z denotes softmax function, and κ is a constant. The C&W algorithm is divided into ℓ_0, ℓ_2, ℓ_∞ depending on the norms.

— ℓ_0 means gradually finding pixel points that have less influence on the classification result and then fixing these pixels. Until no more such unaffected pixel points can be found, the remaining pixels are perturbed.
— ℓ_2 allows a balance between the degree and amount of modification:

$$\text{minimize} \left\| \frac{1}{2} \times (\tanh(w) + 1) - x \right\|_2^2 + c \times f\left(\frac{1}{2} \times (\tanh(w) + 1)\right).$$

— ℓ_∞ restrictions on changes and iterative updates in place of ℓ_2 penalties: minimize $c \times f(x + \delta) + \sum_i [(\delta_i - \tau)^+]$.

In the NLP domain, Sun and Tang [143] provided the C&W algorithm to attack the case prediction model to obtain sensitive content in each doctor-patient record, including information about the patient's medical history and the treatment program proposed by doctors.

UAP. Unlike FGSM, DeepFool, ILCM, which only perturb single images, **Universal Adversarial Perturbation (UAP)** is a special method that perturbs multiple images. Adding universal perturbations on multiple images induces images to be misclassified without solving optimization problems or gradient computation [104]. Specifically, for a d dimensional data distribution μ, there exists a classifier $k(x)$ for each sample $x \in R^d$: $\hat{k}(x + \delta) \neq \hat{k}(x)$ for "most" $x \sim \mu$. The perturbation ξ denotes the radius of the ℓ_p ball and satisfy the constraint $\|\delta\|_p \leq \xi$. $\mathop{\mathbb{P}}_{x\sim\mu}(\hat{k}(x+\delta) \neq \hat{k}(x)) \geq 1-p$. The goal is to identify a perturbation vector δ, which can be added to all examples, and the examples are misclassified with probability $1 - p$. Note that different networks obtain different perturbation results, and even different initializations of the same network do not obtain the same perturbation results. There are other further studies on UAP, such as GD-UAP [107], NAG [125], and AAA [124].

DeepFool. The FGSM needs to select perturbation factor ε manually. To solve this problem, Fawzi and Frossard proposed a more adaptive DeepFool attack [105] and an evaluation metric for classifier robustness. DeepFool does not require selecting a perturbation factor ε and achieves an attack on the general non-linear decision function by multiple linear approximations. A hyperplane can be used to distinguish the two classes for binary and linear classifiers. We only need updates $\boldsymbol{x}$

to the other side of the straight line or hyperplane to obtain $\boldsymbol{x}'$. The projection distance $\frac{f(x_0)}{\|w\|_2^2}w$ of the minimum perturbation θ can be found by measuring the shortest distance from $\boldsymbol{x}$ to the line and making $\boldsymbol{x}$ plus this distance and then converting it to the other side of the plane. Deepfool can perform both untargeted and targeted attacks. The untargeted attack traverses all classes and finds the sample with the least variation, while the targeted attack is performed against the hyperplane of the target class [105].

5.2.2 *Black-Box Attacks.*

ATNs. Most of the traditional adversarial examples are constructed based on gradient information and thus are only suitable for white-box attack scenarios. In contrast, Baluja and Fischer proposed two adversarial example generation methods **Perturbation-ATN** (**P-ATN**) and **Adversarial Autoencoding** (**AAE**) [8] for attacking one or more networks based on **Adversarial Transformation Networks** (**ATNs**). The optimization objective of ATNs is to minimize the joint loss functions L_χ and L_Y to generate adversarial examples: $\arg\min_\theta \sum_{x_i \in \chi} \beta L_\chi(g_{f,\theta}(x_i), x_i) + L_y(f(g_{f,\theta}(x_i)), f(x_i))$. The ATNs can not only perform targeted as well as non-targeted attacks, but also has the option to train the network in a black-box or white-box manner.

ZOO. Inspired by the work of Carlini and Wagner [21], Chen et al. [26] proposed ZOO attack, which generates adversarial examples directly by estimating the gradient of the target model while does not require any information about the model. Chen proposed the ZOO attack by modifying its loss function. ZOO attack is capable of performing both targeted and untargeted attacks. In case of a targeted attack, the loss function is as Equation (5). For an untargeted attack, the loss function can be replaced as Equation (6).

$$f(\mathbf{x}, t) = \max\left\{\max_{i \neq t} \log[F(\mathbf{x})]_i - \log[F(\mathbf{x})]_t, -\kappa\right\}, \tag{8}$$

$$f(\mathbf{x}) = \max\left\{\log[F(\mathbf{x})]_{t_0} - \max_{i \neq t_0} \log[F(\mathbf{x})]_i, -\kappa\right\}. \tag{9}$$

The performance of ZOO and C&W attacks is comparable, but the cost of querying and estimating gradients is higher for ZOO.

Houdini. Houdini is a method that generates adversarial perturbations based on the gradient information of the network's differentiable loss function [29], which can be tailored for the task loss. It is applied not only in image classification but also in speech recognition and semantic segmentation. Houdini can be used both as a non-targeted and targeted attack. In their paper, the authors describe how the Houdini algorithm can be applied in three different domains: Human Pose Estimation, Semantic segmentation, and Speech Recognition. The ABX test results show that it is impossible to distinguish the Houdini-based algorithm using DeepSpeech-2 model [3] to generate the adversarial examples from the original audio.

One Pixel. Su et al. [141] performed an adversarial attack algorithm called one pixel attack that requires only a small number of pixel points to be changed. Only one pixel point needs to be modified in an extreme case, and no information about the network parameters or gradients needs to make the attack. One Pixel Attack also applies to attack models where gradients are not differentiable or difficult to compute. One Pixel Attack does not have many restriction on the magnitude of the perturbations. The perturbations are encoded into matrices, i.e., candidate solutions. Differential evolution is applied to get the optimal solution. A candidate solution contains a fixed number of perturbations. Each perturbation contains five components, x, y coordinates and perturbed RGB. One pixel is modified at a time. Firstly, each pixel is iterated and modified to generate a child image

(candidate solution), subsequently compared with the parent image. Based on the preset determination rules, the child image with the best attack effect is retained accordingly and proceeds to the next iteration. Finally, when the preset number of iterations is reached, or the actual class label is below 5%, the iteration is stopped, and the adversarial attack is completed.

EvadeML. Xu, Qi, and Evans [167] used genetic programming to attack **Portable Document Format** (**PDF**) malware classifiers. The mutation of variants is controlled by the score provided by the Cuckoo sandbox that examines the runtime behavior. In their work, each variant in genetic programming is evaluated by a target classifier as well as an oracle. The target classifier is a black box that outputs the confidence score of malware. Next, the selected variants are randomly manipulated by mutation operators to produce the next generation of the population. The process continues until an evasive sample is found or a threshold number of generations is reached. The authors reported a 100% evasion rate against both the PDFrate and Hidost malware classifier but admitted the process is computationally expensive.

MalGAN. Recently, MalGAN [69] is introduced to generate PE executable malware adversarial examples. This approach creates a fully differentiable substitute model trained to generate modified malware features with corresponding inputs. The substitute model is then used for gradient computation in a modified GAN to produce evasive malware variants. The generator is used to generate adversarial examples. A malware feature vector $\boldsymbol{m}$ and a noise vector $\boldsymbol{z}$ are the inputs to the generator network. Each element in $\boldsymbol{m}$ indicates whether a feature exists or not. The noise vector $\boldsymbol{z}$ is randomly sampled from a uniform distribution $[0, 1)$. Removing features from binary malware during malware feature generation may result in malfunction. Therefore, the authors chose to add irrelevant features instead of removing them. MalGAN does not depend on the model gradient and has a higher success rate of the attack. However, it requires greater overhead compared to the gradient attack approach.

5.2.3 *Other Attacks.*

- Speech recognition. CommanderSong [171] designed the first robust cross-air practical adversarial attack against ASR systems. The researchers considered the speaker noise, receiver distortion, and generic background noise in the physical attack and first built a noise model. Subsequently, noise and random small perturbations were added to the song $X(t)$. The gradient descent optimization was then used to make the pdf-ids of the modified song $X'(t)$ similar to the command b thus fooling Kaldi [117]. Alzantot et al. [2] illustrated the first genetic algorithm-based generation of audio adversarial instances for black-box attacks, thus avoiding the tedious work of computing the MFCC derivatives. The attacker created a population of candidate adversarial examples by injecting random noise and then calculated the fitness score of each population member to generate the next generation of adversarial examples from the current generation by applying selection, crossover, and mutation. Inspired by Alzantot's work [2], Taori et al. [148] crafted targeted adversarial audio based on genetic algorithms and gradient estimation. In order to make it suitable for phrases and sentences, momentum mutation and CTC Loss were introduced. Experiments showed that their research could decode phrases of any length on more complex deep speech systems.
- Natural language processing. Based on Wikipedia reading comprehension examples, Jia and Liang performed ADDSENT and ADDANY to generate sentences that look similar to the question but do not inconsistent in semantics with the correct answer [73]. Song and Shmatikov [136] found that adversarial samples are equally capable of fooling **Optical Character Recognition** (**OCR**). So they generated adversarial images of the printed text of individual words sufficient to cause the Tesseract system to misidentify the subject by

Table 6. Adversarial Examples Defense Methods

Defense motivation	Defense strategy	Defense approach	Advantages	Disadvantages
Model	Modified network	Models Defense distillation [115]	Easy to train, low overhead better generalization, ability	Highly correlated with the model, high computational complexity
		Gradient regularization [106]	Model robustness enhancement	
		Deep compression network [60]	High robustness	
	Model enhancement	Feature squeezing [166]	Low complexity	Impairs model prediction performance
	Additional network	Detector [95]	Low complexity, model irrelevant	Worse generalization, no contribution to model robustness
		GAN-based [74, 82]		
	Validation network	Network verification [56, 75]	High interpretability	Huge computational overheads
Data	Modified training process or input data	Adversarial training [1, 55, 129, 144, 149]	Easy implementation, strong defense capabilities	Difficult to converge, high computational overhead
		Data randomization [163]		
		Data compression [41]		
		Input reconstruction [36, 138]		

replacing the original word with its antonym. The discrete nature of text data hardens the adversarial example attack on NLP tasks. There are still a number of works overcoming such difficultly. To address this challenge, Li et al. [83] proposed Bert-Attack to ensure the semantic consistency and sentence fluency of the generated adversarial samples by replacing words with high semantic impact.

— Malware detection. Gross et al. [59] performed adversarial example attack against malware binary features in Android malware detection. Hu et al. [68] demonstrated an adversarial example sequence generation method that can be used to attack models in RNN detection systems. However, Arjovsky and Bottou [5] pointed out that GAN suffers from training-related stability problems, and it may not converge in a given dataset the attack method in [68] endure such problems as well. Anderson et al. [4] employed reinforcement learning to evade ML-based malware detection for the first time. Specifically, they predefined a series of functionally irrelevant modification operations, and then reinforcement learning is performed to get the optimal sequence of modification operations, but the success rate of the attack was the only 24%.

5.3 Defenses

The adversarial defense technology should avoid excessive modifications to the original model structure while ensuring that the model's performance (e.g., speed, memory usage, model classification accuracy) is not weakened. Methods for defending against adversarial example attacks have been studied extensively in the image classification domain, with speech recognition and malware domains second and NLP domain the least [20]. Therefore, we will elaborate the defense methods against adversarial examples attacks focusing on image classification and a few other domains. We summarize defense methods in Table 6.

5.3.1 Defensive Distillation. Hinton et al. [65] reported the first systematic study of distillation. The idea behind distillation is to migrate fine-grained knowledge from large-scale training models to small-scale models, allowing small-scale to perform learning tasks accurately and more efficiently. Based on Hinton's research, Papernot et al. [115] approved a defensive distillation method. A small model (distilled network) is provided to simulate a large, computationally intensive model (initial) to address the problem of missing information without compromising accuracy. Defensive distillation is capable of defending against most adversarial example attacks and is easy to train. As shown in Figure 13, the specific defense process is as follows: firstly, the teacher model/initial network is trained using the hard label, setting the temperature T. The output class probability of the softmax layer of the initial network is $F(X) = [\frac{e^{z_i(X)/T}}{\sum_{l=0}^{N-1} e^{z_l(X)/T}}]_{i \in 0 \ldots N-1}$. Next, the cross-entropy

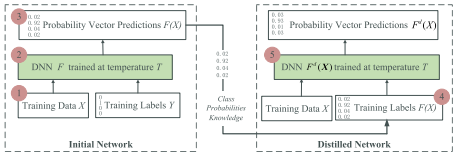

Fig. 13. Architecture of defensive distillation [115].

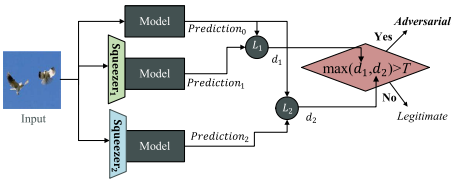

Fig. 14. Feature-squeezing framework for detecting adversarial examples [166].

loss function is calculated. The soft label derived from the teacher model is then utilized for training the distilled network (student model). Finally, the last layer of the distilled network (student model) is modified, and set temperature $T = 1$, thus predicting the class of unknown inputs with high confidence. The advantages of defensive distillation are high generalization ability and low training overhead. However, it does not contribute to the robustness of neural networks [19, 21].

5.3.2 Gradient Regularization. Gradient regularization refers to adding constraints to the objective function during training to avoid significant changes in the model output with changes in the input. Typically, small perturbations do not significantly affect the output. Lyu et al. [21] trained the model using a joint set of regularization methods to defend FGSM-based attacks. It is worth noting that adversarial training significantly reduces the curvature of loss functions and the classifier's decision boundary. Accordingly, Moosavi-Dezfooli et al. [106] proposed a new curvature regularization strategy that directly minimizes the curvature of the loss surface. Their approach significantly improves the robustness of the neural network but may damage the model's performance (e.g., reduce the accuracy). Besides, regularization methods and adversarial training can be combined to defend against adversarial attacks, but the computational complexity is too high.

5.3.3 Deep Compression Networks. Gu and Rigazio constructed **Deep Contractive Networks** (**DCN**) based on **Contractive Auto Encoders** (**CAE**) and proved that the method effectively improves the robustness of neural networks [60]. They trained **Denoising Autoencoders** (**DAE**) to remove the adversarial noise. The objective function of the DCN consists of the autoencoder and a smoothing penalty, which is an end-to-end training procedure. $J_{DCN}(\theta) = \sum_{i=1}^{m}(L(t^{(i)}, y^{(i)}) + \sum_{j=1}^{H+1} \lambda_j \times \| \frac{\partial h_j^{(i)}}{\partial h_{j-1}^{(i)}} \|_2)$, where $t^{(i)}$ and $y^{(i)}$ denote the real and predicted labels of the input $x^{(i)}$, respectively. The penalty term consists of the scaling factor λ and the Frobenius norm $\| \frac{\partial h_j^{(i)}}{\partial h_{j-1}^{(i)}} \|_2$.

5.3.4 Feature Squeezing. There are two methods of data compression in the field of image classification: (i) Reduce the color depth and encode colors with lower values. (ii) Use the spatial smoothing filter, which allows multiple inputs to be mapped into a single value. Feature squeezing is a model-enhancing technology that reduces the representation data's complexity by compressing the input features to resist adversarial perturbations. As shown in Figure 14, Xu et al. [166] processed the input images using the two compression methods mentioned above and then fed them to the same model separately to obtain two prediction results prediction_1 and prediction_2. Finally, prediction_1 and prediction_2 are compared with the prediction_0 of the original unprocessed input to obtain the differences d_1, d_2, respectively. Once this difference is greater than a given threshold T, the input is discriminated as an adversarial example but not legitimate. The effectiveness of the defense varies for different forms of attacks and different squeeze technology. For attacks based on ℓ_0-norm and ℓ_∞-norm are more suitable for the color bit depth squeeze, while for attacks based on ℓ_2-norm, either squeeze method is less effective. It is important to stress that feature squeezing

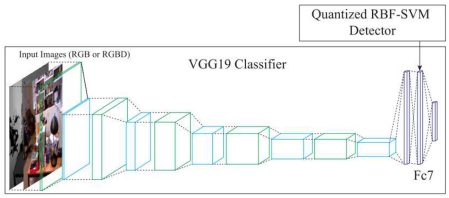

Fig. 15. SafetyNet consists of a conventional classifier with an RBFSVM [95].

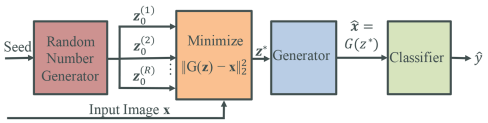

Fig. 16. Overview of the defense-GAN algorithm [128].

does not modify the model itself, so it can be easily integrated with other defense strategies to obtain better defense results.

5.3.5 Detectors. A detector is a mechanism that discriminates whether a detected image is an adversarial sample or not. Usually, the criteria for the detector to discriminate the adversarial samples can be freely defined. The most straightforward approach is to label the adversarial and legitimate samples to train a classifier. There are two main types of classifier training methods. One approach is to train a classifier by directly labeling the adversarial examples and the original samples separately in the initial stage. Another way is to train a classifier by labeling the adversarial and clean samples only on a specific layer's output values. Metzen et al. [103] completed the training of a binary classification task based on a small detector for distinguishing real data from adversarial examples. Li et al. [84] summarized various types of adversarial attacks in the malware detection domain and trained an SVM classifier to defend against adversarial example attacks using a manipulated dataset. As shown in Figure 15, Lu et al. [95] developed SafetyNet, in which the output of the Relu function is used as features of the detector and distinguishes between legitimate samples and adversarial examples by an RBF-SVM classifier. SafetyNet consists of the original classifier (VGG19/ResNet) and the adversary detector (RBF-SVM), which rejects the samples if the detector shows them adversarial. Besides, researchers based on criteria such as PCA [87], maximum mean discrepancy [58], uncertainty [46] to distinguish adversarial examples.

5.3.6 Gan-Based. Attributing to the rise of GAN and its excellent performance [128], many researchers have applied it to adversarial example defense [74, 82]. Samangouei et al. [128] reduced the efficiency of adversarial perturbations based on GAN and proposed a strategy Defense-GAN, a strategy capable of defending against both white-box and black-box attacks. As shown in Figure 16, the Defense-GAN does not modify the classifier structure and training process. However it is crucial to train a stable GAN network and to choose the appropriate hyperparameters, otherwise defense-GAN does not achieve the expected defense. The specific process is as follows:

(1) Generate R random noise vectors $\mathbf{z}_0^{(1)}, \mathbf{z}_0^{(2)}, \ldots, \mathbf{z}_0^{(R)}$ based on random number generators.
(2) Fed the random vector $\mathbf{z}_0^{(1)}, \mathbf{z}_0^{(2)}, \ldots, \mathbf{z}_0^{(R)}$, and the sample $\mathbf{x}$ to the trained GAN network to find $\mathbf{z}^*$ that satisfies the objective function: Minimize $\|G(\mathbf{z}) - \mathbf{x}\|_2^2$.
(3) $\mathbf{z}^*$ is fed into the generator network for training until the generator generates an image $\hat{\mathbf{x}}$ that satisfies the distribution of clean samples.
(4) Classify $\hat{\mathbf{x}} = G(\mathbf{z}^*)$.

5.3.7 Network Verification. Network validation can be used to check whether a sample violates certain properties of DNNs or whether a sample within a determined range (distance from the original sample) changes its label value. In the defense phase of adversarial example attacks, an adversarial example is discriminated if its input is detected to violate some properties of DNNs.

Therefore, network validation is based on the detection of new unknown attacks by the model itself. Katz et al. [75] devised the Reluplex detection method based on the satisfiability modulo theory solver to verify the Relu activation function of a neural network. The high computational complexity of neural network verification is an NP-complete problem, so Katz speeded up the verification by prioritizing the order of the nodes to be checked and sharing information about the verification to speed up the computation. Reluplex only checks the robustness in the vicinity of a few individual input points. Gopinath et al. [56] extended the reluplex method and proposed Deepsafe. It automatically identifies and verifies a safe region of the input space in which the network is robust to specific label misclassification and can defend against specific targeted attacks.

5.3.8 Adversarial Training. As the name implies, adversarial training means adding adversarial examples crafted by attack algorithms such as FGSM to the training set during the training phase. It is a brute force defense scheme and a regularization tool to alleviate the model's overfitting problem. Adversarial training requires vast amounts of adversarial examples to train networks against single-step attacks but ineffective against iterative attacks. The first studies of adversarial training came from Goodfellow et al. [55], who obtained a more robust model in this way. It also means that the ability of the model to defend against adversarial examples are improved. They found that this approach could defend well against white-box attacks but was not useful in black-box scenarios. Tramer et al. [149] further carried out ensemble adversarial training to expand the training set and thus increase the diversity of adversarial samples, achieving a better defense effect even in black box scenarios. In the audio domain, Sun et al. [104] first proposed to train robust acoustic models on natural samples supplemented with adversarial examples, which were dynamically generated by FGSM. They validated the method on two different datasets: Aurora-4, CHiME-4 finding that adversarial training effectively improved the convolutional neural network robustness. Nevertheless, regardless of the defense, new adversarial examples were always found to mount adversarial attacks [104].

5.3.9 Data Randomization. The term data randomization refers to perform randomization technology to mitigate the adversarial effects. Xie et al. [163] mitigated the adversarial effects by randomly resizing the images and using random padding technology to disrupt the structure of specific adversarial perturbations during the forward propagation phase of the model. Specifically, it consists of two steps. (i) Firstly, the input image is resized randomly. (ii) Then, zero random padding is applied around the resized image. The location of the padding is chosen randomly. Data randomization is effective in defending against both single-step attacks and iterative attacks. The data random defense method not only requires no additional training and is less computationally intensive but is also compatible with other adversarial defense methods.

5.3.10 Input Reconstruction. Song et al. [138] found significant differences in the distribution of log-likehoods between perturbed and benign images. In the image training set distribution, the probability density of the adversarial examples was much lower than that of the benign samples and was mainly located in the low probability region. Based on this, Song et al. [138] proposed the PixelDefend defense method to purify the input data to move it back to the high probability region of the training distribution by means of reconstruction to purify the maliciously adversarial images. Specifically, for an input image $\mathbf{X}$, the goal is to find the image $\mathbf{X}^*$ that maximizes the probability of distribution $P(\mathbf{X})$ within the ϵ_{defend} -ball of $\mathbf{X}$. ϵ_{defend} is used to control the example reconstruction behavior. If no adversarial examples are detected, the samples are not changed, $\epsilon_{\text{defend}} = 0$. The PixelDefend defense method does not need to retrain the model, but may not be applicable to scenarios with large spatial distribution of datasets.

5.3.11 Other Defenses. There is a wide variety of methods for generating adversarial examples, and it is difficult to use one defense technology to resist all adversarial example attacks. Therefore, many researchers have started to study the combination of various defense technologies to enhance the effectiveness of defenses. For example, Meng et al. [101] established the MagNet defense framework, which combines one or more separate detector networks and a reformer network, capable of defending against black as well as grey-box attacks. A semidefinite relaxation based technology was introduced by Raghunathan et al. [122] to generate certificates. These certificates are then used to train a more robust network against adversarial example attacks. Prakas et al. [118] performed a new integrated defense method by combining pixel deflection and wavelet denoising technology, which uses adaptive soft thresholding in the wavelet domain to smooth the output of the model. This defense method can effectively defend against the latest adversarial attacks. Zhang et al. [176] trained a multi-redundant architecture defense model on the randomized data using **Voltage Over Scaling** (**VOS**) technology to defend against adversarial example attacks. Since their defense method HRAE requires a lot of training, it is more suitable for offline environments. Gradient masking tries to hide the gradient information of the network, thus preventing the attacker from constructing adversarial examples using gradient solving methods. However, gradient masking is difficult to defend against black-box attacks [149].

6 AI SYSTEM INTEGRATION

AI technology is ubiquitous. Although we have discussed the threats and countermeasures of AI-itself, the security issues appear to be more complex when AI is integrated into real-world applications. The security issues are different for different application scenarios, and we should take a global view of the security of AI. This section will explore several security risks in the actual integration phase.

— AI confidentiality. Confidentiality of AI includes data confidentiality and model confidentiality. Confidentiality of AI generally explicitly related to model privacy, though it can also (indirectly) result in security issues [175], such as model inversion [48] and model extraction [150]. Model inversion refers to the inverse analysis of a model to obtain private data based on the mapping relationship between inputs and outputs. Model extraction, for another, is generally understood to mean executing an acceptable number of queries through an API and observing the output results (probabilities or labels) to infer model parameters or extract an approximate model that closely matches the target model. For both types of privacy problems, privacy risks are commonly mitigated using DP-differential privacy [40, 44], homomorphic encryption [52], or model watermarking [153]. It appears that current AI security and privacy issues are addressed separately. Though challenging, it is worth considering addressing them systematically and concurrently [72, 137], which ensures data and model privacy while maintaining the security of AI systems.

Federated learning [98], proposed by a Google research team, is a new distributed ML technique that has become another critical new branch of AI. Federated learning aggregates local models trained over localized data held by each client to update the global model. Federated learning primarily mitigates privacy issues, but it lacks auditing of local data and control of participant behavior, which is likely to introduce security issues. The client-side of federated learning, central server, and communication channels are easy targets for attackers. The most typical is poisoning attacks, which can be achieved through data poisoning or/and model poisoning [7, 11], user-side/server-side GAN attacks [157], and privacy issues in federated learning [63]. Designing countermeasures against security attacks is more challenging than centralized training due to the lack of access to data and limited control over clients.

— Code vulnerability protection. Current AI system technologies (e.g., deep learning) are built on frameworks (e.g., Tensorflow, Caffe, and Torch). These frameworks rely on a variety of base libraries and third-party components that have greatly facilitated the development of AI technologies. However, they are not designed to be flawless and have vulnerabilities [162]. More coretely, a recent study has shown that unaudited third-party code snippet—the code response for loss computation—can implant backdoor into the deep learning model [6]. Although code vulnerabilities are part of the model implementation process, they are also a critical part of the security deployment of AI systems.

7 CHALLENGES AND OPPORTUNITIES

Although the above mentioned attacks and countermeasures have been widely studied, there are still various challenges and opportunities.

— Identify a defense mechanism that can be applied to different kinds of sensor devices. Due to the continuous updating of new data collection devices and tools, there is abundant room for further progress in protecting the security of data during the data collection process. Sensor security issues mainly lie in the design of the hardware and the malicious signal injection performed by attackers on the hardware. On the hardware side, researchers need to modify and verify the physical properties and logical wiring of the sensor. On the software side, researchers need to design reasonable strategies to identify and reject malicious signals during the data collection phase of the sensor, such as the sensor needs to clean the data signal, amplification, and other pre-processing operations. Secondly, sensor malicious signal identification technology generalization is poor. It is urgent to identify a defense mechanism that can be applied to different kinds of sensor devices, making it impossible for attackers to bypass the defense mechanism.
— Enhancing AI model interpretability. Current research on AI systems lacks the interpretability of models mounted. Yet, we struggle to explicitly analyze and explain what security problems occur in AI under what circumstances. As a result, attackers are always able to find the attack surface of AI, while we are unable to identify potential security vulnerabilities prior to their exposure and exploitation by the attackers. We urgently need to strengthen the mathematical validation of AI models and enhance the interpretability of AI.
— Strengthen the dual protection of security and privacy for AI systems. Security and privacy are closely related, and AI systems not only have the aforementioned security challenges but also suffer from privacy risks [175], such as model inversion [48] and model extraction [150]. The current security and privacy issues for AI are addressed separately. Future research on the integration of both privacy and security areas needs to be strengthened [72, 137] to ensure the privacy issues of data and models while maintaining the security of AI systems.
— Holistic Security Assurance. As pointed by Bertino [10], we can use holistic security assurance to mitigate AI security threats. Simply focusing on a specific episode, e.g., adversarial example attack, or data poisoning, is insufficient or ineffective. This could overlook some facts. For example, while it is important to prevent adversarial example attacks on traffic sign images, it is worthwhile to leverage multiple sources of information, such as map marks and radar information, to jointly make decisions, eliminating the risks of a single source. In addition, when integrating the AI application, system security, data security, and software security can all be taken into account to build a holistic assurance process. That is to say, mounting the practices in other security domains, e.g., the data provenance in data security, can efficiently reduce the risk of data poisoning attacks and help build a secure AI system.

8 CONCLUSION

AI has been widely used in various fields, which will lead to a new round of industrial change and drive human society into the era of intelligence. However, AI technology development has not matured. Security risks exist at all stages of the AI system lifecycle. Although numerous countermeasures have been proposed, there are still various challenges that require to be addressed. This article details the progress of security research in data collection, model training, model inference, and integrated applications of AI systems, and reviews related defense technologies. We conclude by summarizing the challenges and opportunities for security issues related to AI systems. Although AI development technology is in full swing, its increasingly prominent security issues also prompt us to be more vigilant to protect its higher, faster, and better development.

REFERENCES

[1] Abdullah Al-Dujaili, Alex Huang, Erik Hemberg, and Una-May O'Reilly. 2018. Adversarial deep learning for robust detection of binary encoded malware. In *Proceedings of the 2018 IEEE Security and Privacy Workshops*. IEEE, 76–82.

[2] Moustafa Alzantot, Bharathan Balaji, and Mani B. Srivastava. 2018. Did you hear that? Adversarial Examples Against Automatic Speech Recognition. *CoRR abs/1801.00554 (2018). arXiv:1801.00554*. http://arxiv.org/abs/1801.00554.

[3] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, Jie Chen, Jingdong Chen, Zhijie Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Ke Ding, Niandong Du, Erich Elsen, Jesse Engel, Weiwei Fang, Linxi Fan, Christopher Fougner, Liang Gao, Caixia Gong, Awni Hannun, Tony Han, Lappi Vaino Johannes, Bing Jiang, Cai Ju, Billy Jun, Patrick LeGresley, Libby Lin, Junjie Liu, Yang Liu, Weigao Li, Xiangang Li, Dongpeng Ma, Sharan Narang, Andrew Ng, Sherjil Ozair, Yiping Peng, Ryan Prenger, Sheng Qian, Zongfeng Quan, Jonathan Raiman, Vinay Rao, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Kavya Srinet, Anuroop Sriram, Haiyuan Tang, Liliang Tang, Chong Wang, Jidong Wang, Kaifu Wang, Yi Wang, Zhijian Wang, Zhiqian Wang, Shuang Wu, Likai Wei, Bo Xiao, Wen Xie, Yan Xie, Dani Yogatama, Bin Yuan, Jun Zhan, and Zhenyao Zhu. 2016. Deep speech 2: End-to-end speech recognition in English and Mandarin. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*. 173–182.

[4] Hyrum S. Anderson, Anant Kharkar, Bobby Filar, and Phil Roth. 2017. Evading machine learning malware detection. *Black Hat* (2017), 1–6.

[5] Martin Arjovsky and Léon Bottou. 2017. Towards Principled Methods for Training Generative Adversarial Networks. arXiv:1701.04862 [stat.ML] https://arxiv.org/abs/1701.04862

[6] Eugene Bagdasaryan and Vitaly Shmatikov. 2020. Blind Backdoors in Deep Learning Models. CoRR abs/2005.03823 (2020). arXiv:2005.03823 https://arxiv.org/abs/2005.03823

[7] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2020. How to backdoor federated learning. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*. PMLR, 2938–2948.

[8] Shumeet Baluja and Ian Fischer. 2017. Adversarial Transformation Networks: Learning to Generate Adversarial Examples. CoRR abs/1703.09387 (2017). arXiv:1703.09387 http://arxiv.org/abs/1703.09387

[9] Mauro Barni, Kassem Kallas, and Benedetta Tondi. 2019. A new backdoor attack in CNNs by training set corruption without label poisoning. In *Proceedings of the 2019 IEEE International Conference on Image Processing*. IEEE, 101–105.

[10] E. Bertino. 2021. Attacks on artificial intelligence [last word]. *IEEE Security & Privacy* 19, 01 (Jan 2021), 103–104. DOI: https://doi.org/10.1109/MSEC.2020.3037619

[11] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. 2019. Analyzing federated learning through an adversarial lens. In *Proceedings of the International Conference on Machine Learning*. PMLR, 634–643.

[12] Bryan Biegel and James F. Kurose. 2016. The National Artificial Intelligence Research and Development Strategic Plan_NSTC and NITRD. White House (2016). https://www.nitrd.gov/pubs/national_ai_rd_strategic_plan.pdf.

[13] Battista Biggio, Igino Corona, Giorgio Fumera, Giorgio Giacinto, and Fabio Roli. 2011. Bagging classifiers for fighting poisoning attacks in adversarial classification tasks. In *Multiple Classifier Systems*. C. Sansone, J. Kittler, and F. Roli (Eds.), Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 6713, Springer, 350–359. DOI: https://doi.org/10.1007/978-3-642-21557-5_37

[14] Battista Biggio, Blaine Nelson, and Pavel Laskov. 2013. Poisoning Attacks against Support Vector Machines. arXiv:1206.6389 [cs.LG].

[15] Battista Biggio and Fabio Roli. 2018. Wild patterns: Ten years after the rise of adversarial machine learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (Toronto, Canada) (CCS'18)*. Association for Computing Machinery, New York, NY, USA, 2154–2156. https://doi.org/10.1145/3243734.3264418

[16] Benjamin Birnbaum, Brian DeRenzi, Abraham D. Flaxman, and Neal Lesh. 2012. Automated quality control for mobile data collection. In *Proceedings of the 2nd ACM Symposium on Computing for Development*. ACM, New York, NY. DOI: https://doi.org/10.1145/2160601.2160603

[17] Jakramate Bootkrajang and Ata Kabán. 2014. Learning kernel logistic regression in the presence of class label noise. *Pattern Recognition* 47, 11 (2014), 3641–3655. DOI: https://doi.org/10.1016/j.patcog.2014.05.007

[18] Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. 2017. Adversarial Patch. CoRR abs/1712.09665 (2017). arXiv:1712.09665 http://arxiv.org/abs/1712.09665

[19] Nicholas Carlini and David A. Wagner. 2016. Defensive Distillation is Not Robust to Adversarial Examples. CoRR abs/1607.04311 (2016). arXiv:1607.04311 http://arxiv.org/abs/1607.04311

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

[21] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *Proceedings of the 2017 IEEE Symposium on Security and Privacy*. IEEE, 39–57.

[22] Nicholas Carlini and David Wagner. 2018. Audio adversarial examples: Targeted attacks on speech-to-text. In *Proceedings of the 2018 IEEE Security and Privacy Workshops*. 1–7.

[23] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. 2018. Adversarial Attacks and Defences: A Survey. CoRR abs/1810.00069 (2018). arXiv:1810.00069 http://arxiv.org/abs/1810.00069

[24] Guillaume M. J.-B. Chaslot, Mark H. M. Winands, H. Jaap van den Herik, Jos W. H. M. Uiterwijk, and Bruno Bouzy. 2008. Progressive strategies for Monte-Carlo tree search. *New Mathematics and Natural Computation* 04, 03 (nov 2008), 343–357. DOI: https://doi.org/10.1142/s1793005708001094

[25] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian M. Molloy, and Biplav Srivastava. 2018. Detecting Backdoor Attacks on Deep Neural Networks by Activation Clustering. CoRR abs/1811.03728 (2018). arXiv:1811.03728 http://arxiv.org/abs/1811.03728

[26] 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*. 15–26.

[27] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. 2017. Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning. CoRR abs/1712.05526 (2017). arXiv:1712.05526 http://arxiv.org/abs/1712.05526

[28] Andreas Christmann and Ingo Steinwart. 2004. On robustness properties of convex risk minimization methods for pattern recognition. *The Journal of Machine Learning Research* 5 (Dec. 2004), 1007–1034.

[29] Moustapha Cisse, Yossi Adi, Natalia Neverova, and Joseph Keshet. 2017. Houdini: Fooling Deep Structured Prediction Models. arXiv:1707.05373.

[30] Camille Cobb, Samuel Sudar, Nicholas Reiter, Richard Anderson, Franziska Roesner, and Tadayoshi Kohno. 2016. Computer security for data collection technologies. In *Proceedings of the Eighth International Conference on Information and Communication Technologies and Development (Ann Arbor, MI, USA) (ICTD'16)*. Association for Computing Machinery, New York, NY, USA, Article 2, 11 pages. https://doi.org/10.1145/2909609.2909660

[31] D. Cockburn and N. R. Jennings. 1996. ARCHON: A distributed artificial intelligence system for industrial applications. In *Foundations of Distributed Artificial Intelligence*. G. M. P. O'Hare and N. R. Jennings (Eds.), Wiley, 319–344

[32] Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve. 2016. Wav2Letter: an End-to-End ConvNet-based Speech Recognition System. CoRR abs/1609.03193 (2016). arXiv:1609.03193 http://arxiv.org/abs/1609.03193.

[33] Jamie Condliffe. 2015. Robotic Surgery Has Been Connected to 144 US Deaths Since 2000. Retrieved October 28, 2021 from https://gizmodo.com/robotic-surgery-has-been-connected-to-144-u-s-deaths-s-1719202166.

[34] Gabriela F. Cretu, Angelos Stavrou, Michael E. Locasto, Salvatore J. Stolfo, and Angelos D. Keromytis. 2008. Casting out demons: Sanitizing training data for anomaly sensors. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy*. IEEE, 81–95.

[35] Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, and Deepak Verma. 2004. Adversarial classification. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 99–108.

[36] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Li Chen, Michael E. Kounavis, and Duen Horng Chau. 2018. Adagio: Interactive experimentation with adversarial attack and defense for audio. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 677–681.

[37] Jeffrey Dastin. 2018. Amazon scraps secret AI recruiting tool that showed bias against women. Retrieved October 28, 2021 from https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G.

[38] Saharnaz Dilmaghani, Matthias R. Brust, Grégoire Danoy, Natalia Cassagnes, Johnatan Pecero, and Pascal Bouvry. 2019. Privacy and security of big data in AI systems: A research and standards perspective. In *Proceedings of the 2019 IEEE International Conference on Big Data*. IEEE, 5737–5743.

[39] 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 Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 9185–9193. DOI: https://doi.org/10.1109/CVPR.2018.00957

[40] Cynthia Dwork. 2006. Differential privacy. In *Automata, Languages and Programming*. M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener (Eds.), Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Vol. 4052, Springer, 1–12. DOI: https://doi.org/10.1007/11787006_1

[41] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M. Roy. 2016. A study of the effect of JPG compression on adversarial images. CoRR abs/1608.00853 (2016). arXiv:1608.00853 http://arxiv.org/abs/1608.00853

[42] Bill Eidson. 2020. MITRE, MICROSOFT, AND 11 OTHER ORGANIZATIONS TAKE ON MACHINE-LEARNING THREATS. Retrieved October 28, 2021 from https://www.mitre.org/publications/project-stories/mitre-microsoft-others-take-on-machine-learning-threats.

[43] Paul Triolo Graham Webster, Rogier Creemers, and Elsa Kania. 2017. A Next Generation Artificial Intelligence Development Plan: China. Retrieved October 28, 2021 from https://www.newamerica.org/cybersecurity-initiative/digichina/blog/full-translation-chinas-new-generation-artificial-intelligence-development-plan-2017/.

[44] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the ACM Conference on Computer and Communications Security*. DOI: https://doi.org/10.1145/2660267.2660348

[45] Ivan Evtimov, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. 2017. Robust Physical-World Attacks on Machine Learning Models. CoRR abs/1707.08945 (2017). arXiv:1707.08945 http://arxiv.org/abs/1707.08945

[46] Reuben Feinman, Ryan R. Curtin, Saurabh Shintre, and Andrew B. Gardner. 2017. Detecting Adversarial Samples from Artifacts. arXiv:1703.00410 [stat.ML].

[47] Financial Stability Board. 2017. Artificial intelligence and machine learning in financial services - market developments and financial stability implications. *Financial Stability Board* 45 (2017).

[48] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. 2014. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *Proceedings of the 23rd USENIX Security Symposium*. 17–32.

[49] Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *Proceedings of the 2018 IEEE Security and Privacy Workshops*. IEEE, 50–56.

[50] Yansong Gao, Bao Gia Doan, Zhi Zhang, Siqi Ma, Jiliang Zhang, Anmin Fu, Surya Nepal, and Hyoungshick Kim. 2020. Backdoor Attacks and Countermeasures on Deep Learning: A Comprehensive Review. CoRR abs/2007.10760 (2020). arXiv:2007.10760 https://arxiv.org/abs/2007.10760

[51] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C. Ranasinghe, and Surya Nepal. 2019. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference*. 113–125.

[52] Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. 2016. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *Proceedings of the International Conference on Machine Learning*. 201–210.

[53] Yuan Gong and Christian Poellabauer. 2017. Crafting Adversarial Examples For Speech Paralinguistics Applications. CoRR abs/1711.03280 (2017). arXiv:1711.03280 http://arxiv.org/abs/1711.03280

[54] Zhitao Gong, Wenlu Wang, Bo Li, Dawn Song, and Wei-Shinn Ku. 2018. Adversarial Texts with Gradient Methods. arXiv:1801.07175 [cs.CL] https://arxiv.org/abs/1801.07175

[55] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *Proceedings of the 3rd International Conference on Learning Representations*. ICLR.

[56] Divya Gopinath, Guy Katz, Corina S. Păsăreanu, and Clark Barrett. 2018. Deepsafe: A data-driven approach for assessing robustness of neural networks. In *Proceedings of the International Symposium on Automated Technology for Verification and Analysis*. Springer, 3–19.

[57] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*. 369–376.

[58] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick D. McDaniel. 2017. On the (Statistical) Detection of Adversarial Examples. CoRR abs/1702.06280 (2017). arXiv:1702.06280 http://arxiv.org/abs/1702.06280

[59] Kathrin Grosse, Nicolas Papernot, Praveen Manoharan, Michael Backes, and Patrick McDaniel. 2017. Adversarial examples for malware detection. In *Proceedings of the European Symposium on Research in Computer Security*. Springer, 62–79.

[60] Shixiang Gu and Luca Rigazio. 2015. Towards Deep Neural Network Architectures Robust to Adversarial Examples. arXiv:1412.5068 [cs.LG] https://arxiv.org/abs/1412.5068
[61] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. CoRR abs/1708.06733 (2017). arXiv:1708.06733 http://arxiv.org/abs/1708.06733
[62] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2019. BadNets: Evaluating backdooring attacks on deep neural networks. *IEEE Access* 7 (2019), 47230–47244. https://doi.org/10.1109/ACCESS.2019.2909068
[63] Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Richard Nock, Giorgio Patrini, Guillaume Smith, and Brian Thorne. 2017. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. CoRR abs/1711.10677 (2017). arXiv:1711.10677 http://arxiv.org/abs/1711.10677
[64] J. Henry Hinnefeld, Peter Cooman, Nat Mammo, and Rupert Deese. 2018. Evaluating Fairness Metrics in the Presence of Dataset Bias. CoRR abs/1809.09245 (2018). arXiv:1809.09245 http://arxiv.org/abs/1809.09245
[65] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. arXiv:1503.02531 [stat.ML] https://arxiv.org/abs/1503.02531
[66] Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural Computation* 18, 7 (2006), 1527–1554. DOI: https://doi.org/10.1162/neco.2006.18.7.1527
[67] Shengshan Hu, Xingcan Shang, Zhan Qin, Minghui Li, Qian Wang, and Cong Wang. 2019. Adversarial examples for automatic speech recognition: Attacks and countermeasures. *IEEE Communications Magazine* 57, 10 (2019), 120–126.
[68] Weiwei Hu and Ying Tan. 2017. Black-Box Attacks against RNN based Malware Detection Algorithms. CoRR abs/1705.08131 (2017). arXiv:1705.08131 http://arxiv.org/abs/1705.08131
[69] Weiwei Hu and Ying Tan. 2017. Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN. CoRR abs/1702.05983 (2017). arXiv:1702.05983 http://arxiv.org/abs/1702.05983
[70] 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 33rd International Conference on Neural Information Processing Systems*. 125–136.
[71] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. 2018. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *Proceedings of the 2018 IEEE Symposium on Security and Privacy*, Vol. 2018-May. 19–35. DOI: https://doi.org/10.1109/SP.2018.00057
[72] 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 2019 ACM SIGSAC Conference on Computer and Communications Security*. 259–274.
[73] Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, 2021–2031.
[74] Guoqing Jin, Shiwei Shen, Dongming Zhang, Feng Dai, and Yongdong Zhang. 2019. APE-GAN: Adversarial perturbation elimination with GAN. In *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 3842–3846.
[75] Guy Katz, Clark Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. 2017. Reluplex: An efficient SMT solver for verifying deep neural networks. In *Proceedings of the International Conference on Computer Aided Verification*. Springer, 97–117.
[76] Bedeuro Kim, Alsharif Abuadbba, Yansong Gao, Yifeng Zheng, Muhammad Ejaz Ahmed, Hyoungshick Kim, and Surya Nepal. 2020. Decamouflage: A Framework to Detect Image-Scaling Attacks on Convolutional Neural Networks. CoRR abs/2010.03735 (2020). arXiv:2010.03735 https://arxiv.org/abs/2010.03735
[77] Pang Wei Koh and Percy Liang. 2020. Understanding Black-box Predictions via Influence Functions. arXiv:1703.04730 [stat.ML] https://arxiv.org/abs/1703.04730
[78] Yehao Kong and Jiliang Zhang. 2020. Adversarial audio: A new information hiding method. In *Proceedings of the Interspeech*, 2287–2291.
[79] Denis Foo Kune, John Backes, Shane S. Clark, Daniel Kramer, Matthew Reynolds, Kevin Fu, Yongdae Kim, and Wenyuan Xu. 2013. Ghost talk: Mitigating EMI signal injection attacks against analog sensors. In *Proceedings of the IEEE Symposium on Security and Privacy*. 145–159. DOI: https://doi.org/10.1109/SP.2013.20
[80] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. CoRR abs/1607.02533 (2016). arXiv:1607.02533 http://arxiv.org/abs/1607.02533
[81] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. 2016. Adversarial machine learning at scale. In *Proceedings of the 5th International Conference on Learning Representations*. ICLR.
[82] Hyeungill Lee, Sungyeob Han, and Jungwoo Lee. 2017. Generative Adversarial Trainer: Defense to Adversarial Perturbations with GAN. CoRR abs/1705.03387 (2017). arXiv:1705.03387 http://arxiv.org/abs/1705.03387
[83] Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial Attack Against BERT Using BERT. CoRR abs/2004.09984 (2020). arXiv:2004.09984 https://arxiv.org/abs/2004.09984

[84] Wenjia Li, Neha Bala, Aemun Ahmar, Fernanda Tovar, Arpit Battu, and Prachi Bambarkar. 2019. A robust malware detection approach for android system against adversarial example attacks. In *Proceedings of the 2019 IEEE 5th International Conference on Collaboration and Internet Computing*. IEEE, 360–365.

[85] Wenjia Li and Houbing Song. 2015. ART: An attack-resistant trust management scheme for securing vehicular ad hoc networks. *IEEE Transactions on Intelligent Transportation Systems* 17, 4 (2015), 960–969.

[86] Wenjia Li, Houbing Song, and Feng Zeng. 2017. Policy-based secure and trustworthy sensing for internet of things in smart cities. *IEEE Internet of Things Journal* 5, 2 (2017), 716–723.

[87] Xin Li and Fuxin Li. 2017. Adversarial examples detection in deep networks with convolutional filter statistics. In *Proceedings of the IEEE International Conference on Computer Vision*. 5764–5772.

[88] Wei Liang, Songyou Xie, Jiahong Cai, Jianbo Xu, Yupeng Hu, Yang Xu, and Meikang Qiu. 2021. Deep neural network security collaborative filtering scheme for service recommendation in intelligent cyber-physical systems. *IEEE Internet of Things Journal* (2021), 1–1. https://doi.org/10.1109/JIOT.2021.3086845

[89] Huaqing Lin, Zheng Yan, Yu Chen, and Lifang Zhang. 2018. A survey on network security-related data collection technologies. *IEEE Access* 6, 1 (2018), 18345–18365.

[90] Chang Liu, Bo Li, Yevgeniy Vorobeychik, and Alina Oprea. 2017. Robust linear regression against training data poisoning. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. 91–102. DOI: https://doi.org/10.1145/3128572.3140447

[91] Gao Liu, Zheng Yan, and Witold Pedrycz. 2018. Data collection for attack detection and security measurement in Mobile Ad Hoc Networks: A survey. *Journal of Network and Computer Applications* 105 (2018), 105–122. https://doi.org/10.1016/j.jnca.2018.01.004

[92] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2018. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *Proceedings of the International Symposium on Research in Attacks, Intrusions, and Defenses*. Springer, 273–294.

[93] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen Chuan Lee, and Xiangyu Zhang. 2017. Trojaning attack on neural networks. In *Proceedings of the Network and Distributed System Security Symposium*.

[94] Auranuch Lorsakul and Jackrit Suthakorn. 2007. Traffic sign recognition using neural network on OpenCV: Toward intelligent vehicle/driver assistance system. In *Proceedings of the 4th International Conference on Ubiquitous Robots and Ambient Intelligence*, 1–19. Retrieved from http://crit2007.bartlab.org/Dr.Jackrit'sPapers/ney/1.TRAFFIC{_}SIGN{_}Lorsakul{_}ISR.pdf.

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

[96] Nitin Madnani and Bonnie J. Dorr. 2010. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics* 36, 3 (2010), 341–387. DOI: https://doi.org/10.1162/coli_a_00002

[97] John McCarthy. 1956. Artificial Intelligence (AI) Coined at Dartmouth. Retrieved October 28, 2021 from https://250.dartmouth.edu/highlights/artificial-intelligence-ai-coined-dartmouth.

[98] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the Artificial Intelligence and Statistics*. PMLR, 1273–1282.

[99] Shike Mei and Xiaojin Zhu. 2015. Using machine teaching to identify optimal training-set attacks on machine learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 29.

[100] Gys Albertus Marthinus Meiring and Hermanus Carel Myburgh. 2015. A review of intelligent driving style analysis systems and related artificial intelligence algorithms. *Sensors* 15, 12 (Dec 2015), 30653–30682. DOI: https://doi.org/10.3390/s151229822

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

[102] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. 2020. PULSE: Self-supervised photo upsampling via latent space exploration of generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2437–2445.

[103] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. 2017. On Detecting Adversarial Perturbations. arXiv:1702.04267 [stat.ML] https://arxiv.org/abs/1702.04267

[104] 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*. 1765–1773.

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

[106] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. 2019. Robustness via curvature regularization, and vice versa. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9078–9086.

[107] Konda Reddy Mopuri, Aditya Ganeshan, and R. Venkatesh Babu. 2018. Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 10 (2018), 2452–2465.

[108] Lindasalwa Muda, Mumtaj Begam, and I. Elamvazuthi. 2010. Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic TimeWarping (DTW) Techniques. CoRR abs/1003.4083 (2010). arXiv:1003.4083 http://arxiv.org/abs/1003.4083

[109] 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.

[110] Blaine Nelson, Marco Barreno, Fuching Jack Chi, Anthony D. Joseph, Benjamin I. P. Rubinstein, Udam Saini, Charles Sutton, J. D. Tygar, and Kai Xia. 2008. Exploiting machine learning to subvert your spam filter. In *Proceedings of the 1st USENIX Workshop on Large-Scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More*.

[111] Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Emre Kıcıman. 2019. Social Data: Biases, Methodological Pitfalls, and Ethical Boundaries. *Frontiers in Big Data* 2 (2019), 13. https://doi.org/10.3389/fdata.2019.00013

[112] Douglas O'Shaughnessy. 2008. Automatic speech recognition: History, methods and challenges. *Pattern Recognition* 41, 10 (2008), 2965–2979.

[113] Mesut Ozdag. 2018. Adversarial Attacks and Defenses Against Deep Neural Networks: A Survey. *Procedia Computer Science* 140 (2018), 152–161. https://doi.org/10.1016/j.procs.2018.10.315 Cyber Physical Systems and Deep Learning Chicago, Illinois November 5-7, 2018.

[114] Nicolas Papernot, Patrick Mcdaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. 2016. The limitations of deep learning in adversarial settings. In *Proceedings of the 2016 IEEE European Symposium on Security and Privacy*. DOI: https://doi.org/10.1109/EuroSP.2016.36

[115] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016. Distillation as a defense to adversarial perturbations against deep neural networks. In *Proceedings of the 2016 IEEE Symposium on Security and Privacy*. IEEE, 582–597.

[116] George Philipp and Jaime G. Carbonell. 2018. The Nonlinearity Coefficient - Predicting Overfitting in Deep Neural Networks. *CoRR abs/1806.00179 (2018)*. arXiv:1806.00179 http://arxiv.org/abs/1806.00179

[117] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The Kaldi speech recognition toolkit. In *Proceedings of the IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society.

[118] Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. 2018. Deflecting adversarial attacks with pixel deflection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8571–8580.

[119] Katyanna Quach. 2020. Researchers made an OpenAI GPT-3 medical chatbot as an experiment. It told a mock patient to kill themselves. Retrieved October 28, 2021 from https://www.theregister.com/2020/10/28/gpt3_medical_chatbot_experiment/.

[120] Erwin Quiring, David Klein, Daniel Arp, Martin Johns, and Konrad Rieck. 2020. adversarial preprocessing: Understanding and preventing image-scaling attacks in machine learning. In *Proceedings of the 29th USENIX Security Symposium*, 1363–1380.

[121] Erwin Quiring and Konrad Rieck. 2020. Backdooring and Poisoning Neural Networks with Image-Scaling Attacks. *CoRR abs/2003.08633 (2020)*. arXiv:2003.08633 https://arxiv.org/abs/2003.08633

[122] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. 2018. Certified Defenses against Adversarial Examples. *CoRR abs/1801.09344 (2018)*. arXiv:1801.09344 http://arxiv.org/abs/1801.09344

[123] Kanishka Rao, Haşim Sak, and Rohit Prabhavalkar. 2017. Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer. In *Proceedings of the 2017 IEEE Automatic Speech Recognition and Understanding Workshop*. 193–199.

[124] Konda Reddy Mopuri, Phani Krishna Uppala, and R. Venkatesh Babu. 2018. Ask, acquire, and attack: Data-free UAP generation using class impressions. In *Proceedings of the European Conference on Computer Vision*. 19–34.

[125] Konda Reddy Mopuri, Utkarsh Ojha, Utsav Garg, and R. Venkatesh Babu. 2018. NAG: Network for adversary generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 742–751.

[126] Mohsen Rezvani, Aleksandar Ignjatovic, Elisa Bertino, and Sanjay Jha. 2014. Secure data aggregation technique for wireless sensor networks in the presence of collusion attacks. *IEEE Transactions on Dependable and Secure Computing* 12, 1 (2014), 98–110.

[127] Benjamin I. P. Rubinstein, Blaine Nelson, Ling Huang, Anthony D. Joseph, Shing Hon Lau, Satish Rao, Nina Taft, and J. D. Tygar. 2009. Antidote: Understanding and defending against poisoning of anomaly detectors. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference*. DOI: https://doi.org/10.1145/1644893.1644895

[128] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. 2018. Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models. *CoRR abs/1805.06605 (2018).* arXiv:1805.06605 http://arxiv.org/abs/1805.06605

[129] Motoki Sato, Jun Suzuki, Hiroyuki Shindo, and Yuji Matsumoto. 2018. Interpretable Adversarial Perturbation in Input Embedding Space for Text. *CoRR abs/1805.02917 (2018).* arXiv:1805.02917 http://arxiv.org/abs/1805.02917

[130] 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. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 6103–6113.

[131] Hocheol Shin, Yunmok Son, Youngseok Park, Yujin Kwon, and Yongdae Kim. 2016. Sampling race: Bypassing timing-based analog active sensor spoofing detection on analog-digital systems. In *Proceedings of the 10th USENIX Workshop on Offensive Technologies*.

[132] Yasser Shoukry, Paul Martin, Paulo Tabuada, and Mani Srivastava. 2013. Non-invasive spoofing attacks for anti-lock braking systems. In *Cryptographic Hardware and Embedded Systems*. G. Bertoni and J. S. Coron (Eds.), Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 8086, Springer, 55–72. DOI: https://doi.org/10.1007/978-3-642-40349-1-4

[133] Yasser Shoukry, Paul Martin, Yair Yona, Suhas Diggavi, and Mani Srivastava. 2015. PyCRA : Physical challenge-response authentication for active sensors under spoofing attacks categories and subject descriptors. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 1004–1015.

[134] Marco della Cava. 2018. Uber self-driving car kills Arizona pedestrian, realizing worst fears of the new tech. Retrieved October 28, 2021 from https://www.usatoday.com/story/tech/2018/03/19/uber-self-driving-car-kills-arizona-woman/438473002/.

[135] Yunmok Son, Hocheol Shin, Dongkwan Kim, Youngseok Park, Juhwan Noh, Kibum Choi, Jungwoo Choi, and Yongdae Kim. 2015. Rocking drones with intentional sound noise on gyroscopic sensors. In *Proceedings of the 24th USENIX Security Symposium*.

[136] Congzheng Song and Vitaly Shmatikov. 2018. Fooling OCR Systems withAdversarial Text Images. *CoRR abs/1802.05385 (2018).* arXiv:1802.05385 http://arxiv.org/abs/1802.05385

[137] Liwei Song, Reza Shokri, and Prateek Mittal. 2019. Privacy risks of securing machine learning models against adversarial examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 241–257.

[138] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. 2017. PixelDefend: Leveraging Generative Models to Understand and Defend against Adversarial Examples. *CoRR abs/1710.10766 (2017).* arXiv:1710.10766 http://arxiv.org/abs/1710.10766

[139] Jacob Steinhardt, Pang Wei Koh, and Percy Liang. 2017. Certified defenses for data poisoning attacks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Vol. 2017-Decem, 3518–3530.

[140] Carsten Stephan, Henning Cammann, Axel Semjonow, Eleftherios P. Diamandis, Leon F. A. Wymenga, Michael Lein, Pranav Sinha, Stefan A. Loening, and Klaus Jung. 2002. Multicenter evaluation of an artificial neural network to increase the prostate cancer detection rate and reduce unnecessary biopsies. *Clinical Chemistry* 48, 8 (2002), 1279–1287. DOI: https://doi.org/10.1093/clinchem/48.8.1279

[141] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. 2019. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation* 23, 5 (2019), 828–841.

[142] Lichao Sun, Ji Wang, Philip S. Yu, and Bo Li. 2018. Adversarial Attack and Defense on Graph Data: A Survey. *CoRR abs/1812.10528 (2018).* arXiv:1812.10528 http://arxiv.org/abs/1812.10528

[143] Mengying Sun, Fengyi Tang, Jinfeng Yi, Fei Wang, and Jiayu Zhou. 2018. Identify susceptible locations in medical records via adversarial attacks on deep predictive models. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. DOI: https://doi.org/10.1145/3219819.3219909

[144] Sining Sun, Ching-Feng Yeh, Mari Ostendorf, Mei-Yuh Hwang, and Lei Xie. 2018. Training Augmentation with Adversarial Examples for Robust Speech Recognition. *CoRR abs/1806.02782 (2018).* arXiv:1806.02782 http://arxiv.org/abs/1806.02782

[145] Latanya Sweeney. 2000. Simple demographics often identify people uniquely. *Health* 671, 2000 (2000), 1–34.

[146] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *Proceedings of the 2nd International Conference on Learning Representations*. 1–10.

[147] Yasmin Tadjdeh. 2017. DARPA's 'AI next' program bearing fruit. *NDIA's Business & Technology Magazine*. Retrieved from https://www.nationaldefensemagazine.org/articles/2019/7/2/algorithmic-warfare-darpas-ai-next-program-bearing-fruit.

[148] Rohan Taori, Amog Kamsetty, Brenton Chu, and Nikita Vemuri. 2018. Targeted Adversarial Examples for Black Box Audio Systems. *CoRR abs/1805.07820 (2018).* arXiv:1805.07820 http://arxiv.org/abs/1805.07820

[149] 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, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018*. OpenReview.net. https://openreview.net/forum?id=rkZvSe-RZ.

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

[151] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. 2019. Label-Consistent Backdoor Attacks. arXiv:1912.02771 [stat.ML] https://arxiv.org/abs/1912.02771

[152] Daniele Ucci, Leonardo Aniello, and Roberto Baldoni. 2019. Survey of machine learning techniques for malware analysis. *Computers & Security* 81 (2019), 123–147. https://doi.org/10.1016/j.cose.2018.11.001

[153] Ashish Venugopal, Jakob Uszkoreit, David Talbot, Franz J. Och, and Juri Ganitkevitch. 2011. Watermarking the outputs of structured prediction with an application in statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1363–1372.

[154] James Vincent. 2020. What a machine learning tool that turns Obama white can (and can't) tell us about AI bias. Retrieved October 28, 2021 from https://www.theverge.com/21298762/face-depixelizer-ai-machine-learning-toolpulse-stylegan-obama-bias.

[155] Putra Wanda and Huang Jin Jie. 2020. DeepProfile: Finding fake profile in online social network using dynamic CNN. *Journal of Information Security and Applications* 52 (2020), 102465. https://doi.org/10.1016/j.jisa.2020.102465

[156] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. 2019. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy*. IEEE, 707–723.

[157] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. 2019. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *Proceedings of the IEEE Conference on Computer Communications*. IEEE, 2512–2520.

[158] Guanchao Wen, Yupeng Hu, Chen Jiang, Na Cao, and Zheng Qin. 2017. A image texture and BP neural network basec malicious files detection technique for cloud storage systems. In *Proceedings of the 2017 IEEE Conference on Computer Communications Workshops*. IEEE, 426–431.

[159] Fei Xia and Ruishan Liu. 2016. Adversarial examples generation and defense based on generative adversarial network. *arXiv preprint arXiv:1712.00170* (2016).

[160] Huang Xiao, Battista Biggio, Gavin Brown, Giorgio Fumera, Claudia Eckert, and Fabio Roli. 2015. Is feature selection secure against training data poisoning? In *Proceedings of the International Conference on Machine Learning*. 1689–1698.

[161] Qixue Xiao, Yufei Chen, Chao Shen, Yu Chen, and Kang Li. 2019. Seeing is not believing: Camouflage attacks on image scaling algorithms. In *Proceedings of the 28th USENIX Security Symposium*, 443–460.

[162] Qixue Xiao, Kang Li, Deyue Zhang, and Weilin Xu. 2018. Security risks in deep learning implementations. In *Proceedings of the 2018 IEEE Security and Privacy Workshops*. IEEE, 123–128.

[163] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. 2017. Mitigating adversarial effects through randomization. *CoRR abs/1711.01991 (2017)*. arXiv:1711.01991 http://arxiv.org/abs/1711.01991

[164] Guibiao Xu, Zheng Cao, Bao Gang Hu, and Jose C. Principe. 2017. Robust support vector machines based on the rescaled hinge loss function. *Pattern Recognition* 100, 63 (2017), 139–148. DOI: https://doi.org/10.1016/j.patcog.2016.09.045

[165] Han Xu, Yao Ma, Hao-Chen Liu, Debayan Deb, Hui Liu, Ji-Liang Tang, and Anil K. Jain. 2020. Adversarial attacks and defenses in images, graphs and text: A review. *International Journal of Automation and Computing* 17, 2 (2020), 151–178.

[166] Weilin Xu, David Evans, and Yanjun Qi. 2017. Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. *CoRR abs/1704.01155 (2017)*. arXiv:1704.01155 http://arxiv.org/abs/1704.01155

[167] Weilin Xu, Yanjun Qi, and David Evans. 2016. Automatically evading classifiers. In *Proceedings of the 2016 Network and Distributed Systems Symposium*. Vol. 10.

[168] Hiromu Yakura and Jun Sakuma. 2018. Robust Audio Adversarial Example for a Physical Attack. *CoRR abs/1810.11793 (2018)*. arXiv:1810.11793 http://arxiv.org/abs/1810.11793

[169] Chaofei Yang, Qing Wu, Hai Li, and Yiran Chen. 2017. Generative Poisoning Attack Method Against Neural Networks. *CoRR abs/1703.01340 (2017)*. arXiv:1703.01340 http://arxiv.org/abs/1703.01340.

[170] Yanfang Ye, Tao Li, Donald Adjeroh, and S. Sitharama Iyengar. 2017. A survey on malware detection using data mining techniques. *ACM Computing Surveys* 50, 3 (2017), 1–40.

[171] Xuejing Yuan, Yuxuan Chen, Yue Zhao, Yunhui Long, Xiaokang Liu, Kai Chen, Shengzhi Zhang, Heqing Huang, XiaoFeng Wang, and Carl A. Gunter. 2018. Commandersong: A systematic approach for practical adversarial voice recognition. In *Proceedings of the 27th USENIX Security Symposium*. 49–64.

[172] Chenyue Zhang, Wenjia Li, Yuansheng Luo, and Yupeng Hu. 2020. AIT: An AI-enabled trust management system for vehicular networks using blockchain technology. *IEEE Internet of Things Journal* 8, 5 (2020), 3157–3169.
[173] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. 2017. Dolphinattack: Inaudible voice commands. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 103–117.
[174] Jiliang Zhang and Chen Li. 2020. Adversarial examples: Opportunities and challenges. *IEEE Transactions on Neural Networks and Learning Systems* 31, 7 (2020), 2578–2593. DOI: https://doi.org/10.1109/TNNLS.2019.2933524
[175] Jiliang Zhang, Chen Li, Jing Ye, and Gang Qu. 2020. Privacy threats and protection in machine learning. In *Proceedings of the 2020 on Great Lakes Symposium on VLSI*. 531–536.
[176] Jiliang Zhang, Shuang Peng, Yupeng Hu, Fei Peng, Wei Hu, Jinmei Lai, Jing Ye, and Xiangqi Wang. 2020. HRAE: Hardware-assisted randomization against adversarial example attacks. In *Proceedings of the 2020 IEEE 29th Asian Test Symposium*. IEEE, 1–6.

Received February 2021; revised September 2021; accepted September 2021