Almutairi and Barnawi
Journal of Engineering and Applied Science (2023) 70:16
https://doi.org/10.1186/s44147-023-00184-x

Journal of Engineering and Applied Science

REVIEWS Open Access

Securing DNN for smart vehicles: an overview of adversarial attacks, defenses, and frameworks

Suzan Almutairi* and Ahmed Barnawi

*Correspondence: skhwitimalmutairi@stu.kau.edu.sa

Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

Abstract

Recently, many applications have begun to employ deep neural networks (DNN), such as image recognition and safety-critical applications, for more accurate results. One of the most important critical applications of DNNs is in smart autonomous vehicles. The operative principles of autonomous vehicles depend heavily on their ability to collect data from the environment via integrated sensors, then employ DNN classification to interpret them and make operative decisions. The security and the reliability of DNNs raise many challenges and concerns for researchers. One of those challenges currently in the research domain is the threat of adversarial attacks on DNNs. In this survey, we present state-of-the-art research on DNN frameworks, adversarial attacks, and defenses. We discuss each work along with its advantages and limitations and present our thoughts on and future directions for adversarial attacks and defenses.

Keywords: Autonomous vehicles, DNN, Adversarial attacks, Defense

Introduction

Deep learning algorithms such as deep neural networks (DNNs) [1] have emerged in the last decade for many applications, such as image recognition and voice recognition. The accuracy results (high prediction scores) and low false positive rates generated by these models have encouraged researchers to apply DNNs to safety-critical applications such as autonomous vehicles, malware detection and face recognition. Today, industries are beginning to develop autonomous vehicles, or self-driving vehicles, that do not require human intervention [2]. Safety is one of the main concerns in designing self-driving vehicles. Autonomous vehicles bring many benefits, such as saving time, increasing human safety, reducing traffic congestion, reducing the release of carbon, reducing death rates, and decreasing fuel consumption [1, 2]. For example, IHS Markit forecasts that by 2040, autonomous vehicle sales will ex ceed 33 million. Society of Automotive Engineers (SAE) International introduced six levels for classifying automated systems in vehicles, ranging from level 0 (no au tomation) to level 5 (full automation). Level 4 (high automation) is in the market and is providing services such as Google Waymo [3] and TuSimple [4] to the public, but level 5 (full automation) is still under testing [5].

The safety of autonomous vehicles' driving depends on the robustness of their DNN classifiers. DNN classifiers depend heavily on sensors for high-accuracy object detection. Correct interpretation leads to correct decisions [6]. This means that DNN classifiers have to be impervious to any small modifications in input images that could otherwise lead them to misclassify objects [7]. This raises many concerns and challenges for researchers regarding DNN reliability and security. One of the most serious security issues in DNNs is the threat of adversarial attacks, also known as adversarial examples.

The main goal of adversarial DNN attacks is to use DNN vulnerabilities and generate an adversarial image capable of fooling DNNs into producing incorrect predictions [8–11]. An adversarial attack in an image classification model is considered successful if the generated adversarial image is classified by the target DNN as a different class label (not the correct image class) with a high confidence rate [12]. The following equation expresses this mathematically in simple form: $\bar{x}=x+\epsilon$.

Many attack strategies have been developed to fool DNN models in various domains and applications. Examples of these attacks are DeepSearch [13], greedy local search [12], the Fast Gradient Sign Method (FGSM) [14] and Projected Gradient Descent (PGD) [7]. In addition, various recent studies have proposed a number of defenses for increasing the security and robustness of DNN models [15–18]. Increasing DNN robustness and protection against adversarial attacks is a growing research challenge.

The potential risks associated with the DNN classifiers mentioned above will affect the development of autonomous vehicles and their deployment in industry. If autonomous vehicles cannot ensure human safety on the road, consumers will not accept this technology. Therefore, it is essential to determine whether deep learning systems in autonomous vehicles are vulnerable, how they could be attacked, how much damage could be caused by such attacks and what measures have been proposed to defend against these attacks. The industry needs this analysis and information to improve the safety and robustness of DNNs.

The motivation for conducting this survey was the realization that the world may someday depend on autonomous vehicles to make life easier [17]. However, autonomous vehicles have experienced high rates of accidents compared with human-driven vehicles, though these accidents involve fewer injuries. According to the National Law Review, there are 9.1 autonomous vehicle accidents per million miles driven on average [19]. The main reason for this high rate is the frequency of at tacks on DNN classifier systems in autonomous vehicles [20].

Several researchers, including [7, 12, 21], have recently presented survey papers concerning adversarial attacks on autonomous vehicles. These attacks can be digital or physical. The aim of digital attacks is to find image pixels that generate new fake images, while physical attacks aim to change the environments where the autonomous vehicles exist. Deng et al. [12], provided a survey investigating various types of digital adversarial attack techniques. Examples of digital attacks include the Iterative targeted fast gradient sign method (IT-FGSM) [22], optimization universal adversarial perturbation (Opt uni) [23], AdvGAN [24], and AdvGAN uni [24]. Another survey, conducted by Modas et al. [7], discusses attacks and countermeasures using physical adversarial attacks on autonomous vehicles. Examples of physical attacks are patch or sticker attacks [25], ultrasonic attacks [26, 27] and lidar attacks [28]. However, there is still a lack of systematic surveys

on DNN adversarial attack and defense techniques and DNN behavioral tests in the digital and physical autonomous vehicle environment. In addition, there is an urgent need to combine these techniques into one survey to guide researchers for future improvement. Therefore, this survey on DNN adversarial examples aims to reveal potential threats in autonomous vehicles' physical and digital environment to encourage researchers to deploy defense techniques in advance. Also, artificial intelligence security research has become an important research direction. The contributions of this survey are as follows:

- This survey summarizes the presented generation algorithm to formalize DNN adversarial attacks in autonomous vehicles' digital and physical environment. Also, how to apply the DNN adversarial attacks generated in the digital environment to the physical environment is also discussed.
- We investigate and discuss the latest critical defense techniques against DNN adversarial attacks in the autonomous vehicle environment, provide descriptions of these techniques, and explain the main observations of these methods.
- We provide taxonomies for DNN adversarial attacks, defenses and DNN behavioral tests to systematically analyze these techniques.
- We discuss the issues with the existing research on DNN behavioral tests, adversarial attacks and defenses, and, based on this, recommend future work.

In the following sections, we give an overview on autonomous vehicles and deep learning algorithms and their roles in autonomous vehicle technology. Next, we present DNN vulnerabilities. Then, we discuss the adversarial attacks taxonomy as well as the DNN defense taxonomy and DNN testing frameworks. Finally, we identify and present the challenges and opportunities for future work on DNN adversarial attacks, defenses, and testing frameworks.

Main text

The remainder of this paper is organized as follows: "DNN in Autonomous Vehicles" section introduces the aspects of autonomous vehicles and the background needed to understand the DNN concepts discussed throughout this paper. "Adversarial attack taxonomy" section presents the adversarial attack taxonomy. The "Defenses taxonomy" section appears in section 4. "DNN evaluation framework" section presents the criteria that have been used to test and evaluate DNNs. "Discussion and future research directions" section discusses challenges and potential directions for future research on adversarial attacks. Finally, "Conclusions" section concludes the paper.

DNN in autonomous vehicles

This section presents information on autonomous vehicle technology, the background information needed to understand the deep learning concepts and reinforcement learning.

Autonomous vehicles

Autonomous vehicles employ artificial intelligence techniques such as intelligent agents. An agent perceives its environment through sensors, decides on suitable actions and

applies these actions in the environment through actuators. Automated driving technology consists of three basic functional layers: a sensing layer, a perception layer and a control (decision) layer. The sensing layer has many types of sensors, including lidar, camera, and radar sensors. These sensors are located in the front and back of the vehicle. Lidar sensors are used for many purposes, such as object detection, during which the sensor detects light waves reflected by the objects in the surrounding environment. Camera sensors are used to capture video of the surrounding environment and images such as road signs. Radar sensors are used for simulating vision, monitoring weather conditions and avoiding road objects. The sensors are responsible for collecting data from the environment. The perception layer contains DNNs to analyze and interpret the data collected by the sensors and extract logical information, as shown in Fig. 1. This layer takes the input signals and processes them to make sense of the information they provide the system. The decision layer is responsible for decision-making, such as routing, and controls driving, self-parking, determination of the steering angle and lane detection [1, 29].

SAE International has introduced six levels for classifying automated systems in vehicles. These levels range from level 0 (no automation) to level 5 (full automation). Table 1 presents an overview of these levels and the criteria that distinguish them. For more information about automated vehicles, see [29].

Deep neural networks overview

Deep learning is a subset of machine learning in which the abstraction of the underlying knowledge is learned from the dataset. The main difference between machine algorithms and DNNs is that DNNs do not require domain knowledge or feature engineering because they are end-to-end learning processes. This learning process, which is referred to as representation learning, makes it more transferable to other models. Deep learning algorithms use multiple layers to learn data features. Moreover, algorithms require immense datasets to learn and process because the efficiency of DNNs depends on the

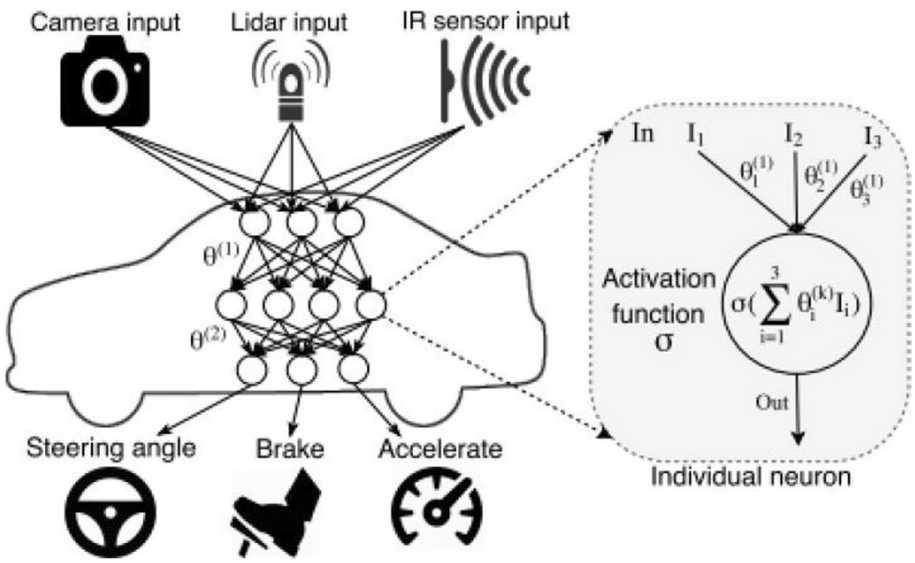

Fig. 1 Overview of the DNN role in an autonomous driving model [12]

Table 1 The six levels of vehicle automation systems

Level	Name	Controller	Require human monitoring	Driving features	Example of features
0	No automation	Human	Yes	Support	Automatic emergency braking, blind spot warning, lane departure warning
1	Driver assistance	Human and system	Yes	Support	Lane centering or adaptive cruise control
2	Partial automation	System	Yes	Support	Simultaneous lane centering and adaptive cruise control
3	Conditional automation	System	No	Automated	Traffic jam chauffeur
4	High automation	System	No	Automated	Features can drive the car in limited conditions
5	Full automation	System	No	Automated	Features can drive the car everywhere in all conditions

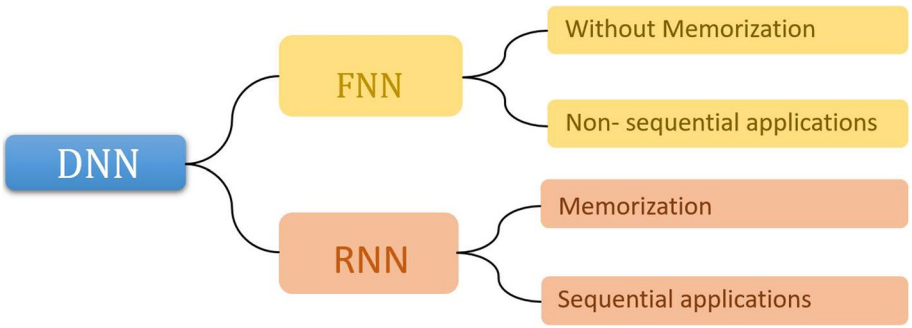

Fig. 2 The difference between RNNs and FNNs

dataset size and vast computer resources. DNNs in autonomous vehicles can be classified into two main types [30]:

1 Feed-forward neural networks (FNNs)
2 Recurrent neural networks (RNNs)

The main difference between feed-forward neural networks (FNNs) and recurrent neural networks (RNNs), as shown in Fig. 2, is that FNNs do not retain the values of the last layer neurons, which means that the neuron values propagate in one direction. This feed-forward operation makes FNNs more suitable for non-sequential applications, such as images. RNNs, meanwhile, memorize the output of the last layer of neurons, which renders them suitable for sequential applications, such as audio [31].

FNN

Convolutional neural networks (CNNs) are examples of FNNs. The CNN architecture has two parts: fully connected neural layers and convolutional layers. Fully connected

neural networks are connected networks of layers. The formal architecture consists of an input layer, one or more hidden layers and an output layer. The input layer is responsible for passing the input data to the hidden layers. The hidden layer or layers are responsible for extracting the features and for information analysis. The output layer is responsible for predicting the input class. For example, as shown in Fig. 3, each neuron (blue circle) in any layer L_i is connected to every neuron in the next layer L_{i+1} (red edges). This structure applies to each neuron in each layer, except the last layer (output). The hidden layers are denoted in Fig. 3 as h1 and h2. The circles represent neurons, and the edges represent the inner product between the corresponding weights and the previous layer neurons. If the result of this multiplication is high, the information or feature relevant to this neuron is important and the neuron should be activated. The convolution layers have neurons that are connected only to certain neurons in the next layer, and the same weights are shared among different connections for different neurons [32].

RNN

The architecture of RNNs is shown in Fig. 4. In addition to the operation process in CNNs, RNNs allow the memorized last neuron output to be considered in the calculation of the current neuron prediction; that is, the neuron output is fed to the previous layer and the next layer. This operation determines the prediction output for the previous input. This approach is helpful for many applications, such as video frame sequences, where the current frame is predicted by the previous frame [30].

Reinforcement learning

The agent in reinforcement learning (RL) learns how to improve its behavior in a certain environment by interacting with that environment. Unlike supervised learning, the relation and mapping from the input to the output is not told explicitly to the agent. Rather than using trial and error, a reward function is used to evaluate actions and update

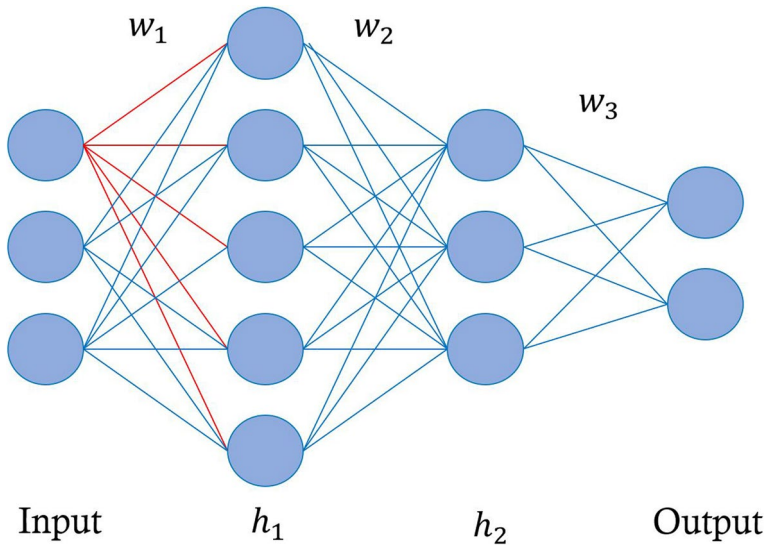

Fig. 3 The simple architecture of fully connected layers

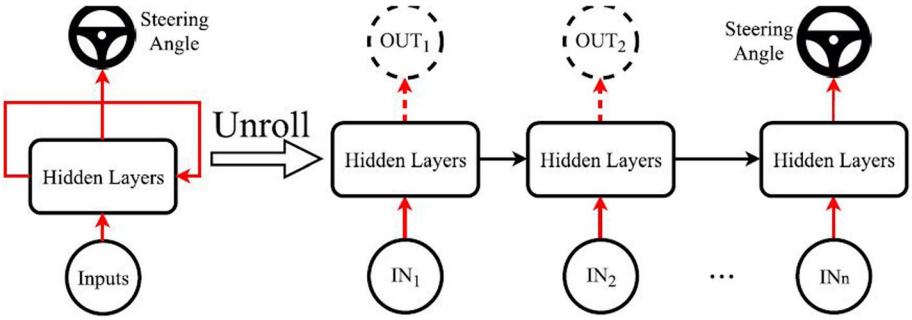

Fig. 4 The basic architecture of RNNs [30]

performance [33–35]. There are several DRL approaches for autonomous vehicles' decision making in adversarial settings such as [36, 37]. A framework was proposed by [38] to evaluate an adversarial agent based on DRL. The aim was to measure the reliability of autonomous vehicles' mechanisms for avoiding collisions and motion planning. Another study [39] proposed a defense that is an extension of two game-theoretic algorithms (robust adversarial reinforcement learning and neural FSP) to a semi-competitive game environment. However, for the next sections we will concentrate on DNN.

Adversarial attack taxonomy

In this section, we first present DNN adversarial vulnerabilities and then discuss adversarial image generation methods. Finally, we present the adversary's means, the adversary's goals, and the adversary's knowledge along with studies on these topics.

DNN adversarial vulnerabilities

DNN adversarial image appearance was first demonstrated by Goodfellow [14] in 2014. Many vulnerabilities exist in DNNs, leading to adversarial attacks. Below, we discuss some of these vulnerabilities.

DNN decision boundary vulnerability

The basic layout of DNNs is to take a raw image as input and output the correct classification label. The classification can be either a binary label or a multiclass label. DNNs consist of a hidden layer with weights and an activation function that can recognize the underlying object structure. As mentioned in "DNN adversarial vulnerabilities" section, DNNs are end-to-end learning processes. This characteristic opens the door for adversaries to exploit DNN vulnerabilities and generate new methods of attack [40].

To understand adversarial attacks on DNNs, consider Fig. 5, which shows a binary model that can classify the input as an orange region (class 1) or a yellow region (class 2). The distribution of the data corresponds to these two classes. The cross points (x) correspond to the images used to train the DNN model. The red line corresponds to the decision boundary learned during the training phase to produce the final classification label.

The decision boundary means that the images under the line will be predicted as class 1 and that those above the line will be predicted as class 2. As shown in the figure, the decision boundary does not fill all the data to avoid an overfitting problem, which causes

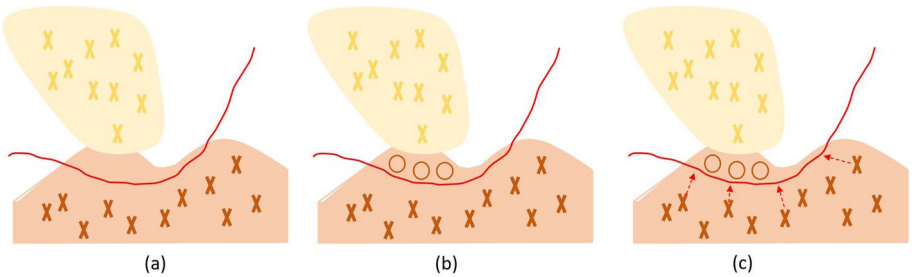

Fig. 5 **a** The nature of DNN training data with decision boundary, **b** adversarial point within orange class but crossing DNN decision boundary, **c** moving the point of the input image from point x to point O to fool the DNN model into misclassifying the input image

the DNN model to predict well with the training data but poorly with the test data. The adversary takes this limitation of the DNN learned decision boundary and obtains a small perturbation value (point O) that falls within the orange region but crosses the decision boundary of the DNN model. The adversary will try to move point (x) across the decision boundary to reach point (O). Thus, the adversary starts to fool the model into misclassifying the points as class 2, where they actually exist in class 1. Therefore, one of the reasons for adversarial image appearance is the decision boundary vulnerability of the trained DNN model [18]. To explain further, in the distribution of the data in the trained model, where each class has a region and boundary, the decision boundary limits the distance between the data within the same class. Thus, that area needs to be maintained so that data cannot easily move to other areas [28].

DNN transferability vulnerability

Another reason for adversarial attack appearance is the DNN transferability property [41]. Consider two models, model A and model B, with the same domain and classification. We can generate adversarial images from model A and poison model B with these data to lead model B to misclassify the data [42, 43]. Research has shown that in this way, vulnerability can be transferred from an insecure model to a secure model [8, 9, 14, 44].

To build robust DNN models that resist adversarial attacks, we need to understand the reasons behind adversarial images and fully understand the structure of DNNs.

Adversarial image generation methods

In this subsection, we discuss two main ways attackers modify original images to produce adversarial images: image distance metric and image transformation.

Image distance metric

Consider that we have a classifier in model G and a clean sample image x with its true label y. The adversary goal is to find and generate a synthesized image $\bar{x}$ that looks perceptually similar to x but can still lead the classifier to misclassify the image as a different, incorrect class t [40]. The most widely employed norms and their meanings are shown in Table 2 [9, 51, 52]. Various studies have been conducted using various norms, such as the one-pixel attack [53], to demonstrate how to constrain the L_0 norm to limit

Table 2 Most popular distance norm used in adversarial images

Symbol	Meaning	Examples
*L*0	Total number of pixels that differ between clean and adversarial image	[45–47]
*L*2	Squared differences between pixel values of clean and adversarial image	[47–49]
L∞	Maximum pixel difference between clean and adversarial image	[14, 47, 50]

the number of pixels that can be changed in the clean image. The allowed perturbation in this case is one pixel, which generates an adversarial image.

Image transformation

Adversarial images can be generated through one of two methods of image transformation [30, 40, 54]:

1 Image pixel transformation focuses on changing the pixel value of the original (clean) image. An example of image pixel transformation is changing the pixel color depth and brightness (referred to as a semantic attack). Traditional adversarial attack generation focuses on changing the image similarity metric to a metric less than the L_p norm, while the newest attacks, which are referred to as semantic attacks, focus on changing the image similarity metric to a metric greater than the L_p norm [9, 52].
2 Image affine transformation, which focuses on spatial modification. Examples of image affine transformation are rotation, translation, and scaling.

Threat model The information about various kinds of adversarial attacks is dependent on the various adversary capabilities. This means that each adversarial attack can be classified by the adversary's means, goal or knowledge about the target DNN. As shown in Fig. 6, we present the adversarial threat model, which organizes the attacks that target DNN models. We discuss these categories in in more detail in the next subsections.

Adversary means

Adversarial attacks can be generated using physical modifications or digital modifications, as shown in Fig. 7.

Physical attack

In a physical attack, the perturbator changes the environment of the application domain. For example, in autonomous vehicles, the adversary may change a stop sign to a 45-mph speed limit sign [25]. The physical modification can be effected using items such as stickers and printed posters. Fooling the DNN and misleading the model can lead to incorrect decisions such as ignoring a stop sign, causing accidents and compromising the safety of humans [17]. Another type of adversarial attack on autonomous vehicles can be accomplished through modified camera angles, such that, in the training set, the image is stored with a fixed degree [55]. The DNN can recognize the motorbike correctly, but

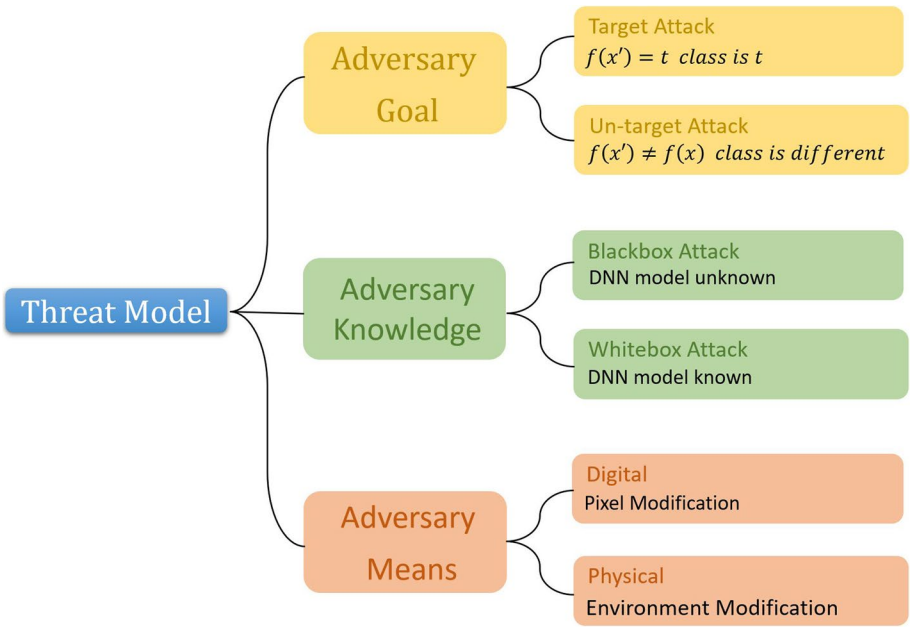

Fig. 6 Adversarial threat model

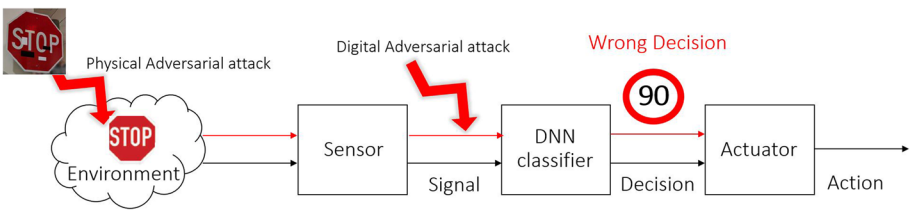

Fig. 7 Digital and physical adversarial attacks

only after changing and rotating the image. The camera viewing angle has been modified according to the DNN, which causes the motorbike to be misclassified as a person [56].

Digital attack

In a digital attack, perturbation is employed after the DNN's weaknesses are explored by altering some of the predefined pixels of the input image. The purpose is to modify the input image in such a way that the model will misclassify it [7]. An adversarial attack in an image classification model is considered successful if the generated adversarial image is classified by the target DNN as a different class (not the correct image class) with a high confidence rate [12].

Adversary goal

Based on the adversary's goals, an adversarial attack can be targeted or non-targeted.

Targeted attack

In a targeted attack, the adversary's goal is to force the DNN model to change the correct class label of an input image to a specific, different target class label.

Non-targeted attack

In a non-targeted attack, the adversary's goal is to force the DNN model to change the correct class label to any other class label [40].

Adversary knowledge

An adversarial attack can be categorized as one of two types, based on the adversary's knowledge: a white-box attack or a black-box attack.

Black-box attacks

In a black-box attack, the structure of the deep neural network is not known to the adversary. The algorithm parameters are unknown as well. This makes the attack more challenging for the adversary. The adversary tries to estimate the gradients of the target DNN model in order to produce an adversarial image. The adversary can access the output of the model and query the target model to obtain the probability scores of all classes. Examples of these attacks are Zeroth-Order Optimization (ZOO) [57], DeepSearch [13], and greedy local search [58].

One of the most famous classical strategies in black-box attacks is ZOO. In this type of attack [57], the adversary completes multiple forward passes on the target model to estimate the gradient. As shown in Eq. 1, the forward pass f is performed on the clean image x and a small perturbation ϵ_i. This equation determines the probability score of the logits of the model.

$$\frac{\partial f(x)}{\partial x} = \frac{f(x + h\epsilon_i) - f(x - h\epsilon_i)}{2h} \tag{1}$$

Amin et al. [52] proposed a shadow attack that targets certified defenses. The shadow attack was successful at breaking randomized smoothing [59] and crown interval-bound propagation [60] defenses. This shadow attack is the generalization of the PGD attack. It concentrates on generating an adversarial example with a spoofed certificate. The attack algorithm works to change the image brightness or darkness with a small change in color depth. However, this attack design was designed specifically for untargeted attacks. The computational cost was not discussed, and the attack was not tested on road sign images.

Additionally, Hamdi et al. [56] discussed semantic adversarial diagnostic attacks (SADA) that are likely to occur, such as a change in camera viewpoint, lighting conditions or other aspects. Semantic attacks are difficult to understand, diagnose, analyze, and study as investigating real-world semantic attacks is not an easy task. Moreover, generating the parameters of the semantic condition is complicated. Hamdi et al. proposed an algorithm and a general setup for the adversarial attack. This algorithm was designed to learn the underlying distribution of semantic adversarial attacks. The proposed general setup includes an entity called the adversary. The interaction between the agent and the adversary takes place through the environment. The adversary tries to give an input to the environment such that the agent will fail in that environment. Then, the adversary receives a score from the agent so that it can update itself and increase attack rates in the future. This attack can be generated on the dataset of images (pixel) or

semantic parameters. The setup Hamdi et al. created consisted of object detection, self-driving and unmanned aerial vehicle racing. They used a YOLOv3 object detector as the agent of their SADA framework. SADA could be used as an attack scheme as well as a diagnostic tool to assess the systematic failure of agents. However, this attack focuses on 2D images within neural networks.

White-box attacks

In a white-box attack, the adversary targets aspects of the DNN such as the structure, parameters and gradient descent. The adversary in this type of attack can obtain all the information needed to build an adversarial attack able to fool the target system [40, 42]. Examples of these attacks are FGSM [14], PGD [7, 61], Carlini, and Wagner (CW) attacks [9] and Jacobian-based saliency map attacks [46].

One of the most famous classical strategies in white-box attacks is FGSM. In this type of attack [14], the adversary computes an adversarial image by adding a pixel perturbation of the magnitude in the direction of the gradient. This computation is done as shown in Eq. 2: the model takes the gradient of the loss with respect to the input image x. Then, it finds the sign and adds an ϵ in the direction of that sign to the input image and generates an adversarial image. The goal is to add a perturbation that increases the loss. The attack then requires one update to achieve an untargeted attack (an adversarial image).

$$x_{\text{adv}} = x + \varepsilon.\,\text{sign}\left(\nabla\, {}_{x}Lx, y_{\text{true}}\right) \tag{2}$$

In a targeted attack, the procedure is the same. However, the ϵ is added in the negative direction of the gradient of the input image x. It is also assumed here that the gradient is computed with respect to the target label y_{target}, as shown in Eq. 3. Hence, the aim here is to reduce the loss to the target label.

$$x_{\text{adv}} = x - \varepsilon.\,\text{sign}\left(\nabla\, {}_{x}Lx, y_{\text{target}}\right) \tag{3}$$

The parameters in Eqs. 2 and 3 are as follows: x is the clean input image, x_{adv} is the corresponding adversarial image, L is model loss, y_{true} is the actual label, y_{target} is the target label and ϵ is in L_∞ budget. FGSM is a single-step method that is efficient in terms of computation time to implement [62]. There is also an iterative version of FGSM [63].

The previously mentioned attacks are performed within the L_p norm and are independent for each sample. However, new attacks aim at finding one perturbation (universal) for all samples of the targeted DNN model [23, 64, 65]. For more information about universal adversarial attacks, see [66–68].

Other strategies exist for performing white-box attacks that do not use the L_p norm to create adversarial attacks. One of these strategies is spatially transformed adversarial attacks [69, 70]. The goal of this type of attack [69] is to find the flow in which the pixels are moved in a certain quantity and perform a bilinear interpolation to ensure that flow does not receive positions that lie between two pixel locations, thus creating an adversarial image.

Pei et al. [71] proposed a White-box tool for DNNs in safety-critical applications, especially for corner situation behavior. This tool is named DeepXplore, and it relies on

the assumption that there are at least two classifiers with the same task but with different dataset training and parameters. Pei et al. introduced a new metric for measuring the number of neurons activated (that meet the DNN classifier rules) by the test input. The proposed approach has two stages: maximizing differential behavior and neuron coverage. The first stage is aimed at obtaining test input that can cause two DNN classifiers to classify the input as different labels. This is achieved by solving the two DNNs' joint optimization problem, which means finding a test input that lies between these DNNs' decision boundaries. The second stage is maximizing the neuron coverage to reach this test input, which is achieved by increasing the number of activated neurons to obtain the test input. Before the application of DeepXplore, the DNNs classified the image with the same label; after the application of DeepXplore, they produced different labels. The framework created by Pei et al. can be utilized to systematically test a real-scenario DL system. However, generating the test input is not an easy task, especially if the DNNs have the same decision boundaries. Moreover, DeepXplore requires that the compared DNNs have the same task. In addition, if three DNNs are compared, DeepXplore guarantees that at least one will be different in label classification. This approach is not effective if the defense algorithm employs majority voting for the final classification score for at least three classifiers.

Likewise, Eykholt et al. [25] proposed a Robust physical Perturbations (RP2) attack algorithm. This algorithm produces perturbations for various physical dynamic environmental conditions encountered by autonomous vehicles. Therefore, the algorithm has been applied to road signs. It has two stages. First, the algorithm identifies the weak area in the image, taking into account various factors such as the image background, camera distance from the image, and camera viewing angle. Then, the algorithm produces a perturbation mask that is printed on white and black stickers. Finally, these stickers are attached to the physical target image in a specific location inconspicuous to the human eye. Their algorithm provides a standardized methodology to evaluate physical adversarial attacks. However, Eykholt et al. did not study the effects of the lighting on the image in the attack success rate. Finally, we summarize the presented studies on adversarial attacks, as shown in Fig. 8. First, based on the adversary's means, we categorize the attack under consideration as physical or digital. Then, based on the adversary's knowledge, we further classify the attack as blackbox or whitebox.

Defenses taxonomy

This section presents the taxonomy of DNN defenses, namely, defense strategies, DNN defense techniques, and DNN detection techniques.

Defenses strategy

Traditional DNN security evaluation methods primarily focus on the accuracy of DNN model classification and fail to evaluate model security and reliability. To address this issue, various recent studies have proposed a number of defenses for increasing the security and robustness of DNN models [15, 16, 72–74]. To evaluate DNN security, two main concepts describe DNN resistance to adversarial attacks:

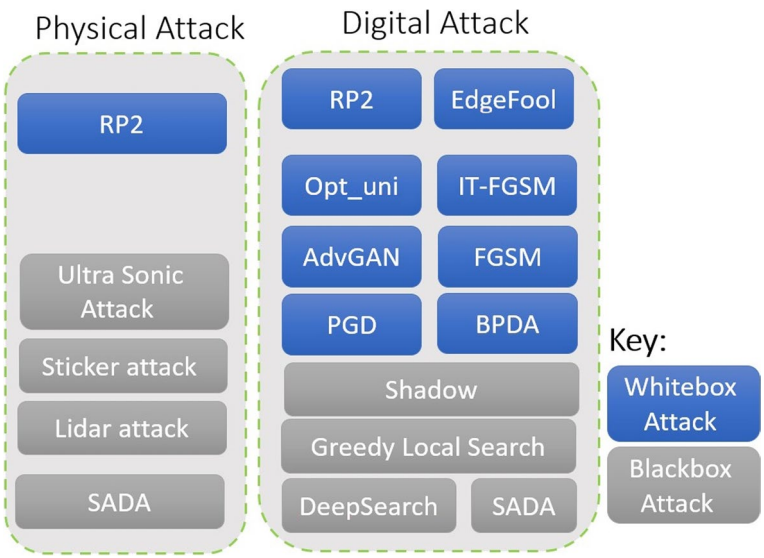

Fig. 8 Taxonomy of adversarial attack studies

1 The first concept is DNN model robustness. This means the DNN model has knowledge of the minimum perturbation that will drive image x to adversarial attack image $\bar{x}$ under this model.
2 The second property is adversarial risk, which refers to the loss function (gradient descent) of the DNN model. In the DNN learning process, the model attempts to increase its prediction score by minimizing error with respect to the input image. Therefore, to create an adversarial image, the adversary attempts to maximize the loss function. This means finding the point in the neighborhood boundaries of x that can fool the DNN model [7].

We can categorize DNN defenses based on their goals when developed against adversarial attacks into two main models:

1 Robust models F(., θ) can correctly classify adversarial attacks [75–78].
2 Robust detection models D(., θ) can detect adversarial attacks [7].

Next, we present the DNN defense models against adversarial attacks in Fig. 9.

DNN defense techniques

Based on the existing research, defenses classified under the first goal can be categorized into five types:

Gradient obfuscation (masking)

As we discuss in "DNN detection techniques" section, white-box attacks attempt to maximize the gradient descent. To counter this approach, the defender can mask the gradient descent in the neural network [75, 79, 80]. This can be accomplished by using a zero gradient, fuzzy gradient or non-existent gradient. However, this type of defense is not effective against black-box attacks.

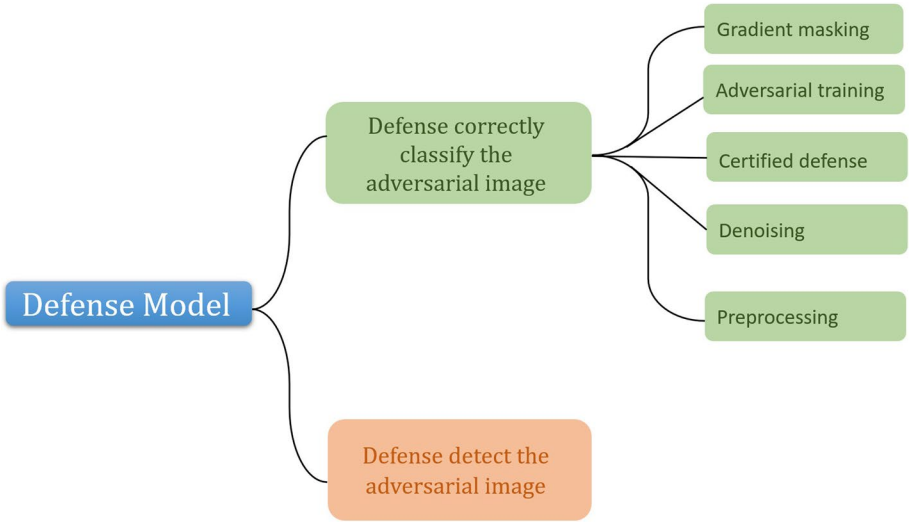

Fig. 9 Adversarial defense model

Hakim [17] has proposed a technique for high-level security. The proposed technique consists of three stages: an ally patch extractor, a CNN evaluator and a final labeling decision. In the first stage, the input image is divided into equally sized ally patch candidates. Then, the generated patch candidates are filtered out using two constraints: minimum text information and the non-redundant patch. The final collective set is fed into the CNN evaluator. In the CNN evaluator, each patch is fed into a separate CNN model where the models are trained on the same task and produce the same label in cases of similarity. There are three scenarios for adversarial patches. In the first scenario, the input image has a large amount of text information and does not pass the filter process in the ally patch extractor stage. In the second scenario, the patch is partial to the intended adversarial patch and is not classified as the adversary target class. In the third scenario, the adversarial patches are assigned to the desired adversary class. In the final labeling decision stage, one of the following mechanisms is performed to make a decision: majority voting, total confidence, weighted average confidence and spanning measure. An ally patch works well on clean or adversarial images. In addition, Hakim determined that the attack success rate was reduced by one-third. However, each of the four strategies used in fusion to produce the final result had nearly the same effect. Moreover, this step is highly time and resource consuming.

Likewise, Wu et al. [15] proposed a defense against 5G-based adversarial attacks on autonomous vehicles. The autonomous camera captures the road image and sends it to Mobile Edge Computing (MEC) using 5G. Then, using Singular Value Decomposition (SVD), certain areas in the captured image are filtered out to eliminate the perturbation, if any. Finally, based on majority voting, the result is returned to the autonomous vehicle for the correct action. Based on this experiment, the tail and the middle area of the image are the most effective areas. In addition, the proposed method is effective against poster-printing, sticker, CW, Deepfool and I-FSGM adversarial attacks. However, the proposed defense's accuracy is affected if the communication switches to 3G or 4G due to signal interference. Moreover, there is no backup solution should 5G or MEC

go down. Finally, when this method was applied, the accuracy of the DNN model on normal images was reduced by 1.75%.

Adversarial training

Adversarial training [7, 76] entails training models with adversarial attacks generated by specific attacks; however, it cannot prevent new attacks. Moreover, it increases network capacity and consumes time, and the model accuracy with a clean dataset may be decreased [81, 82]. AbouKhamis et al. [83] applied the min-max algorithm to the DNN model to investigate its robustness against different adversarial attacks. The min-max algorithm has two parts: the first part aims to obtain an adversarial image from the original image at a high loss gradient, and the second part aims to minimize adversarial loss and increase DNN robustness. However, the researchers used adversarial training as a defense method, which causes a decrease in the classifier accuracy.

Certified defense

Certified defenses [84] check model robustness against attacks. These defenses determine how many samples cannot be attacked in the DNN model. This is done by defining a security parameter ϵ that must be less than the used $L\ p$ norm. Then, an ϵ bounded ball with a radius able to resist identified Lp perturbations is determined around each pixel [52].

Denoiser

The goal of this defense [78, 82, 85] is to remove noise from the adversarial image. The authors of [82] have proposed a denoiser that attempts to minimize the loss function between the output of the original image and the output of the adversarial image.

Sutanto and Lee [78] proposed a defense technique against adversarial attacks based on deep image prior (DIP). This algorithm works by eliminating noise from the adversarial image. The goal of this defense is to construct a noiseless image after various iterations. With each iteration, the parameter is updated in the DIP loss function. These researchers provided a comparison between two images to prove the algorithm's effectiveness. The first image is the original (clean) image minus the adversarial image. The second image is the denoised image (after applying DIP) minus the original. Sutanto and Lee found that the second image had no adversarial image pattern. The experiment results show the effectiveness of DIP against FGSM; however, the number of iterations needed for the denoised image was not discussed. Moreover, the time needed was not discussed, so its potential for effectiveness in critical-safety applications cannot be determined from this study. In addition, the CNN accuracy with the original dataset before applying DIP was 95%, while the accuracy after applying DIP was 90%. This implies that DIP decreases the algorithm's accuracy with clean images.

Hu et al. [86] proposed a denoising process combined with a chaotic encryption defense for adversarial attacks. Their approach works in three stages. First, the input image is encrypted using a discretized baker map. Then, the encrypted input image is passed into a U-net denoiser classifier. Finally, the denoised input image is decrypted. The proposed approach is easy to implement and suitable for high-resolution images.

This approach was applied with encryption and without decryption. The denoised input without encryption was effective against FGSM but failed against PGD attacks. The denoised input with encryption approach was effective against PGD attacks, but the model's accuracy against FGSM and with clean images was reduced. In addition, the proposed approach only works on square image classifiers.

Preprocessing

Preprocessing-based methods include image transformations [77, 87], generative adversarial networks (GANs) [88–90], noise layers [91, 92], denoising auto-encoders [93, 94], and dimensionality reduction [95–97]. The goal of these methods is to perform various transformations on the adversarial image to remove adversarial noise and send the preprocessed image to the target model. The previous studies were evaluated on a small subset of images [98].

Qiu et al. [99] proposed a preprocessing function that is performed on the input sample to remove any adversarial noise before it is fed to the DNN classifier. The proposed approach consists of two major steps. In step one, Discrete Cosine Transform (DCT) is used to transfer the pixels into the frequency coefficients space. Then, these frequency coefficients are quantized with the novel quantization technique. Finally, the result of the previous step is de-quantized, and inverse-DCT is used to transform the pixels back to the spatial space. Step two is used to improve the image distortion as a preprocessing function. This step is novel as it provides a random variance without any influence on the classifier's performance between the clean and the transformed image. The proposed preprocessing step will drop a predefined ratio of image pixels and modify a large amount of pixel coordination. This function provides three security requirements: usability, defensive quantization and approximation difficulty. The proposed approach outperformed comparable defenses such as FD [97], Rand [77], SHTELD [100], TV [87], JPEG [87], BdR [101], and PD [102].

However, the defense is specifically aimed at gradient adversarial attacks.

DNN detection techniques

Defenses classified under this second goal check whether the image is clean or adversarial before feeding the image to the DNN. If the input image is adversarial, it is rejected and does not pass to the DNN classifier. The research focused on this goal is [16, 18, 80]. These approaches identify features that are satisfied by natural (real) images and are not satisfied by adversarial (faked) images. However, this technique is not effective against white-box attacks where the adversary has knowledge about the identified features.

Feature squeezing is an important metric in adversarial image detection. It works by reducing the search space available for an adversary and detecting adversarial images. This can be done by applying transformations such as bit depth and spatial smoothing. These transformation techniques do not change the semantics of real images. The detection model performs two predictions. The first prediction is performed on the input image without any transformation, and the second prediction is performed on the input image after the transformation is applied. Then, the model calculates probability based

on the results of the two predictions and compares it with a specific threshold. If the input image is clean, the two predictions, before and after transformation, will be the same [16].

Li and Velipasalar [18] have proposed a novel weighted average precision (wAP) frame distance metric to detect adversarial objects in autonomous vehicles. The proposed approach has two stages. The first stage is the frame distance metric algorithm, which calculates the differences between the results of two detected object images. Then, based on the frame distance result, the temporal detection score is calculated to determine if this image is adversarial or not. The proposed algorithm focuses on a single frame. Moreover, the proposed algorithm performed better than the existing single-frame detection method. However, experimental results show that the wAP outperformed mean average precision (mAP) in white-box attacks, but wAP and mAP yielded nearly the same results for black-box attacks (Gaussian noise and brightness). In addition, the proposed algorithm cannot be applied to images.

Likewise, Xiao et al. [103] proposed AdvIT, an adversarial detection method for video frames in autonomous vehicles. AdvIT takes the target video frame x_t and reconstructs the optical flows between the x_t frame and its previous frames. AdvIT then generates pseudo frames by transforming the estimated flows with small randomness. AdvIT checks the consistency between frame x_t and the pseudo frames. If the compared frames are consistent, the target frame is clean. AdvIT is the first detection method based on video frames' temporal consistency. In addition, AdvIT performance is not time-consuming. Moreover, AdvIT showed its effectiveness in three video tasks: semantics segmentation, human pose estimation and object detection. However, Xiao et al. compared their method with JPEG. Both methods achieved the same detection rate in semantic segmentation and human pose estimation if $k = 1$, where k was the number of the previous video frames. Moreover, this approach can only detect adversarial attacks and does not provide any defense mechanisms.

Next, we summarize the presented adversarial defense studies in Fig. 10.

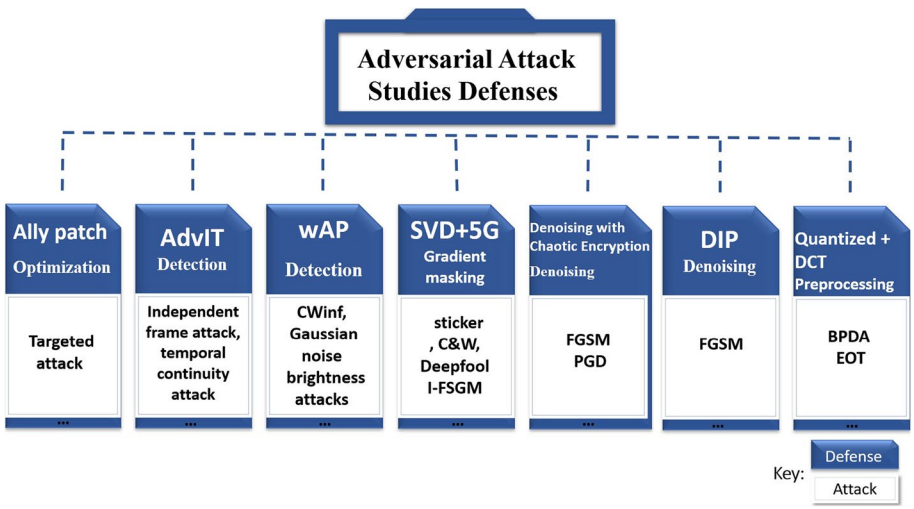

Fig. 10 Adversarial attack defense studies

DNN evaluation framework

Traditional DNN security evaluation methods primarily focus on the accuracy of DNN model classification and fail to evaluate the security and reliability of such models [8, 9, 20]. One way to evaluate the behavior of DNN classifiers is to use behavioral testing to validate the input of the model and the output behavior by performing various tests on system capabilities [30, 54, 104]. Behavioral testing is done without any knowledge about the system's internal structure. The basic idea is to test whether the model behaves in the correct way in various conditions [105]. In traditional program testing, it is preferable to generate more cases to cover all possible cases and detect a code error if it exists. Following the same principle, a systematic method that can generate test input capable of detecting the unexpected/error behavior in DNNs must be established. This problem has been noted by many researchers [20, 30, 54, 106]. It is difficult to generate test data that are representative of the large input data space (dataset), in which various criteria, such as realism and diversity, are met. Moreover, a suitable oracle is needed to explore the entire input-output data space. Oracle effectiveness entails making DNNs generate the correct behavior or problem when the data are fed to the input test. In this technique, it is also challenging for complex domains, such as autonomous vehicles, to identify the correct behavior for every test input [20, 30]. However, oracle can be identified as the relationship between the expected behavior and a certain type of test input. According to Riccio et al., 12 studies have been conducted to trace the oracle issue in machine learning [20].

Like traditional programs, DNN classifiers need to be tested and verified before deployment to the real environment. Two important issues need to be considered in DNN testing: testing criteria, such as neuron coverage [71] and testing strategies, such as coverage-guided testing [30].

Tian et al. [30] proposed a systematic methodology called DeepTest to evaluate DNN classifier behavior in autonomous vehicles. Their method consists of various steps. First, the input-output pair space of the DNN logic is explored through the application of neuron coverage. Second, various image pixel transformations and affine transformations are performed. Third, neuron coverage is increased through the combination of different types of transformation based on the guided greedy search algorithm. Last, a metamorphic to correlate each input to the same output and automatically identify erroneous behavior in the DNN is identified. DeepTest generates test samples that mimic real environmental changes in driving constraints, such as rain or lighting. In addition, this approach focuses on generating test samples for corner cases to detect DNN misbehavior. However, DeepTest cannot guarantee that it will generate a synthetic image that covers all real cases. In addition, DeepTest was designed to test only the steering angle actions taken by autonomous vehicles.

Table 3 present a summary of the existing DNN behavioral tests, attacks, and defenses, including their advantages and limitations.

Discussion and future research directions

The previous subsections discussed various black-box and white-box attacks on autonomous vehicles, different solutions currently proposed for adversarial attack techniques and various studies on DNN behavioral tests. The safety of autonomous

Table 3 Comparison of the existing models of adversarial attacks, defenses, and test frameworks

Category	Year	Strengths	Limitations
Attack [71]	2017	* Their attack succeeded in breaking seven defenses and was published by ICLR2018.	*This attack is not effective against certified defenses and black-box defenses.
Behavior test [30]	2017	* DeepTest generated test samples that mimic real environment changes in driving constraints such as rain or lighting. * They focus on generating test samples for corner cases to detect DNN misbehavior in these cases.	* DeepTest cannot guarantee that it will generate synthetic images that can cover all the real cases. * DeepTest was designed to test only the steering angle action taken by the autonomous vehicles.
Attack [25]	2019	* RP2 provides a standardized methodology for evaluating physical attacks.	* They did not study the effects of the lighting on the image in the attack success rate.
Defense [17]	2019	*Ally patch works well on clean or adversarial images. * The attack success rate was reduced by one-third.	* The four strategies used in fusion to produce the final result showed practically no differences. *Moreover, this step is time- and resource-consuming.
Defense [103]	2019	* AdvIT is the first detection method based on the video frames temporal consistency. * AdvIT performance is not time-consuming. * AdvIT shows its effectiveness in three video tasks: semantics segmentation, human pose estimation, and object detection	* They compare their method with JPEG, where both methods have the same detection rate in semantic segmentation and human pose estimation if $k = 1$, where k is the number of previous video frames. * AdvIT can only detect adversarial attacks and does not provide any defense mechanism.
Attack [56]	2020	*SADA could be used as an attacking scheme as well as a diagnostic tool to assess the systematic failure of agents.	* SADA focuses on 2D images in neural networks.
Attack [52]	2020	* * The shadow attack was successful in breaking randomized smoothing and crown interval bound propagation defenses. * Shadow attack is the generalization of the PGD attack.	* Shadow attack design for untargeted attacks. * The computational cost was not discussed. The attack was not tested on road sign images.
Defense [18]	2020	* The proposed algorithm focuses on a single frame (image). *The proposed algorithm performs better than the existing single-frame detection methods such as mAP.	*The proposed algorithm cannot be applied to images.
Defense [15]	2020	*The tail and the middle area of the image are the most effective areas. It proves its effectiveness against various attacks.	*The defense accuracy is affected if the communication switches to 3G or 4G due to signal interference. *There is no backup solution in case 5G or MEC goes down. *The accuracy of the DNN model on normal images was reduced by 1.75% after applying their method.
Behavior test [104]	2020	* They provide useful metrics for RNN evolution.	* DeepStellar cannot scale for testing large datasets.
Behavior test [54]	2020	*DeepHunter can scale to large datasets, and the metamorphic mutation can generate valid test samples with a validity ratio of 98%. *The seed selection strategy provides diversity and newness of seed selection, which increases neuron coverage and defect detection. *DeepHunter can capture DNN defects during quantization to various platforms.	* The defined threshold for neuron coverage cannot detect fake images.

Table 3 (continued)

Category	Year	Strengths	Limitations
Defense [78]	2020	*The experiment results show the effectiveness of DIP against FGSM.	* The number of iterations needed for the denoised image was not discussed. * The time or the speed needed was not discussed, so its potential for effectiveness in critical-safety applications cannot be determined from their study. * The CNN accuracy on the original dataset before applying DIP is 95%, while the accuracy after applying DIP is 90%. This implies that DIP decreases the algorithm accuracy on clean images.
Defense [99]	2021	*The proposed preproccessing function provides three security requirements: usability, defensive quantization and approximation difficulty. * The proposed approach outperforms the compared defenses.	* The defense is specific for gradient adversarial attacks
Defense [86]	2022	* The proposed approach is easy to implement and suitable for high-resolution images.	* The proposed approach was effective against PGD attack, but the model accuracy against FGSM and clean images was reduced. *The proposed approach can work only on square image classifiers.

Fig. 11 The main challenges

vehicles relies on two important components: DNN classifiers and the sensors that feed these classifiers with captured images. This section first discusses the observations of the previously mentioned information and summarizes the challenges (see Fig. 11). Then, it provides possible suggested future research directions (see Fig. 12) to make autonomous vehicles safer.

Discussion

Trained DNN model errors

Autonomous cars rely on a DNN model perception system to detect objects and drive on their own (without a human driver). However, DNN algorithms have vulnerabilities,

Fig. 12 Future directions

bugs or errors, which may lead the perception system in autonomous cars to misclassify objects and ca used car accidents [3, 4]. One example of such an autonomous car accident involved Uber [107, 108] when an autonomous car that misclassified a pedestrian as a wrong object and failed to prevent a collision. In a safety-critical applications it is important to make sure the trained DNN model for object detection is bug-free and robust.

Autonomous vehicle sensors

An autonomous vehicle without sensors is blind, like a human without eyes. Attacks on autonomous vehicles can be generated against any sensor. The most important sensors in self-driving vehicles are lidar, radar, and camera sensors. Researchers have shown through experiments on SADAs that self-driving vehicle camera sensors are vulnerable to small perturbations in the camera position or camera view angle [56].

Adversarial attack

DNN security and robustness in autonomous vehicles represent one of biggest challenges in this research area [109]. One of the most serious security issues in DNNs is the threat of adversarial attacks. Adversarial attack generation is not an easy task, whether it is a black-box attack or a white-box attack. If the DNN model architecture is complex and masks important model parameters that can be used to generate the attack, designing a white-box attack is difficult. In addition, developing a black-box attack is more difficult. This is because black-box attacks are generated through digital means, such as poising an attack, or physical means, such as modifying the environment with no idea about the classification architecture in the vehicle. Moreover, each defense presented is designed to break a specific attack.

DNN training privacy

We have presented various studies that test DNN classifiers and evaluate their expected behavior; however, there are some limitations, such as scalability and

diversity. Researchers urgently need to find a systematic methodology for testing DNNs in safety-critical applications. The dataset used to test DNN classifiers is an important metric in exploring various unexpected behaviors of DNN classifiers. This can be done by generating input test samples that cover the large input space of deep learning models. This is a difficult goal to accomplish because these samples must meet different criteria such as increasing neuron coverage, diversity and realism [20]. Test input samples can be generated through various methods, such as an adversarial attack. However, adversarial samples will only cover a small subset of the features learned by DNNs as these adversarial samples were not generated to maximize neuron coverage [71].

Trade-off between adversarial accuracy and standard accuracy

Defenses built on adversarial training need to take this topic into consideration. This is because DNN models perform excellently on training data with a high accuracy [78]. After adversarial attacks are generated for the model and the model is retrained on these data, its accuracy on clean images is reduced, while its robustness against attacks is enhanced. Researchers are beginning to investigate this topic [110, 111], but we more work needs to be done on this type of defense to balance the model accuracy.

Future directions

DNN security and robustness

DNN classifiers have to resist any modification that can result in an object or image being misclassified. Therefore, there is a need to develop a systematic approach or test tool to evaluate DNN robustness [25]. One of the ways to do this is to test DNN classifiers against unknown attack scenarios. Likewise, research must be conducted to develop a general technique that can defend against existing attack methods. In addition, the DNN classifier must strongly and accurately detect the object in the real time environment, whether a fake object has been added or a real object has been deleted or modified. Moreover, this defense should be applicable to various types of attacks. For example, adversarial training methods are robust against the attack for which they have been generated. Moreover, this defense must be secure against black-box attacks. It must also have countermeasures in case the DNN comes under attack by the following:

- Isolating the DNN classifier from decisions and waiting for an action from a central authority.
- Updating the system through 5G to double-check final decisions.
- Remotely controlling the vehicle through a central authority.

Toward certified epsilon robust defense

The development of various tests is needed to test DNN classifier robustness in autonomous vehicles. In theory, this can be done if the DNN model in question has a bound

area with respect to the input image. This means that any modification in the input image will result in the same correct classification [112].

Autonomous vehicles' sensor enhancement

Increasing the number of sensors will provide better environment data and increased sensor redundancy, but it will also increase the cost of autonomous vehicles. To counter this system weakness, we suggest the following possibilities:

1. DNN classifiers need to be trained with more quality data and various images that are captured in different positions and from different angles. This means there is a need to build a set of accurate and up-to-date images or videos depicting the autonomous vehicle environment. In the presented studies, researchers applied their algorithms on known datasets with limited classes and images. Thus, there is a need to create new datasets to train DNNs to various scenarios.
2. The number of camera sensors in autonomous vehicles needs to be increased, and they need to be positioned to capture images from various angles. This will increase classifiers' opportunities to make correct predictions. This can be done using multiple DNN classifiers. Each camera sensor will feed an image to a classifier. Then, each classifier will produce a prediction for that image. Finally, based on the final stage components, such as majority voting or weighted average confidence, the final prediction score can be obtained.
3. Lidar sensors can capture objects in 3D point clouds and feed them to DNNs for classification. 3D point cloud models suffer from adversarial attacks as presented in this work [113]. However, these attacks were later shown to be easily rebuffed using simple strategies such as random sampling and denoiser [114]. Moreover, 3D neural network adversarial attacks do not transfer to other unseen 3D networks [114, 115]. Newer studies [43, 115] have shown that 3D neural network adversarial attacks do transfer to other unseen 3D networks with a success rate of 40% and can break existing defenses [115]. Little research has focused on 3D neural network attacks and defenses [43, 114, 115]. Researchers need to move forward to develop attacks and defenses to increase 3D DNN robustness and reliability.

DNN test data generation

This highlights the need to design a good metamorphic mutation strategy including image distance metric and image transformation to generate samples that maintain semantics as close to those of the original sample as possible. The generated samples should be as realistic as possible to mimic various real-world scenarios. This method aims to generate unseen samples for models that help them to analyze and explore different cases. In addition, producing large test samples requires huge manual labeling efforts. To solve this issue, researchers have begun using a metamorphic oracle to facilitate mapping a group of test inputs to the correct behavior (label) and measure whether the test input meets the expected behavior [30, 54].

Conclusions

Deep neural networks (DNNs) are rapidly emerging as a means for classifying images and objects with high accuracy rates. DNNs serve as the foundation for many useful applications, such as facial detection and recognition systems and the safety-critical applications that are of supreme importance for the successful operation of autonomous vehicles. Indeed, one of the most eagerly awaited widespread applica tion domains promised by DNN is that of autonomous vehicles.

Given our growing reliance on DNNs, concerns have been raised regarding their security and reliability. In this survey, we have presented the state-of-the-art research on DNN behavioral tests, adversarial attacks and defenses and have discussed each work with its advantages and limitations. Moreover, we have presented our thoughts on DNN behavioral test adversarial attacks and defenses and have recommended a future direction for this field of study.

This paper concludes that research needs to be carried out regarding general adversarial attacks to develop defenses that are robust against various types of attacks. In addition, a defense that offers a balance between the standard accuracy of DNN models before and after training on adversarial attack samples must be developed. Moreover, researchers must focus on developing models that offer certified robust defenses. Also, research must be performed on increasing the number of autonomous vehicle sensors to provide better environmental data and increased sensor redundancy. Finally, researchers must develop a systematic methodology for evaluating DNNs in autonomous vehicles.

Abbreviations

DNNs	Deep neural networks
SAE	Society of Automotive Engineers
PGD	Projected gradient descent
FGSM	Fast gradient sign method
ITFGSM	Targeted fast gradient sign method
Opt uni	Optimization universal adversarial perturbation
FNNs	Feed-forward neural networks
RNNs	Recurrent neural networks
CNNs	Convolutional neural networks
ZOO	Zeroth-order optimization
CW	Carlini and Wagner attacks
SADA	Semantic adversarial diagnostic attacks
RP2	Robust physical perturbations
GANs	Generative adversarial network
MEC	Mobile edge computing
SVD	Singular value decomposition
DIP	Deep image prior
DCT	Discrete cosine transform
wAP	Weighted average precision
mAP	Mean average precision
PCA	Principal component analysis

Acknowledgements
Not applicable.

Authors' contributions
All authors read and approved the final manuscript.

Funding
This study had no funding from any resource.

Availability of data and materials
Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 17 November 2022 Accepted: 14 February 2023
Published online: 03 March 2023

References

1. Chaitra PG, Deepthi V, Gautami S, Suraj HM, Kumar N (2020) Convolutional neural network based working model of self driving car - a study. In: 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), pp 645–650. https://doi.org/10.1109/ICESC48915.2020.9155826 ID: 1
2. Meftah LH, Braham R (2020) A virtual simulation environment using deep learning for autonomous vehicles obstacle avoidance. In: 2020 IEEE International Conference on Intelligence and Security Informatics (ISI). pp. 1–7. https://doi.org/10.1109/ISI49825.2020.9280513. IEEE.
3. Youn, S.: UPS joins race for future of delivery services by investing in selfdriving trucks. https://abcnews.go.com/Business/ups-joins-race-future-delivery-services-investing-driving/story?id=65014414 Accessed 17 Aug 2019
4. DeBord M (2018) Waymo Has Launched Its Commercial Self-driving Service in Phoenix- and It's Called 'Waymo One'. https://www.businessinsider.com/waymo-one-driverless-car-service-launches-in-phoenix-arizona-2018-12. Accessed 5 Dec 2018
5. Cao Y, Wang N, Xiao C, Yang D, Fang J, Yang R, Chen QA, Liu M, Li B (2021) Invisible for both camera and lidar: Security of multi-sensor fusion based perception in autonomous driving under physical-world attacks. 2021 IEEE Symposium on Security and Privacy (SP). https://doi.org/10.1109/sp40001.2021.00076
6. Liu P, Fu H, Ma H (2021) An end-to-end convolutional network for joint detecting and denoising adversarial perturbations in vehicle classification. Comput Visual Media 7(2):217–227
7. Modas A, Sanchez-Matilla R, Frossard P, Cavallaro A (2020) Toward robust sensing for autonomous vehicles: an adversarial perspective. https://doi.org/10.1109/MSP.2020.2985363 https://ieeexplore.ieee.org/document/9127857
8. Papernot, N., McDaniel, P.D., Goodfellow, I.J.: Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. (2016) CoRR abs/1605.07277. 1605.07277
9. Carlini N, Wagner D (2017) Towards evaluating the robustness of neural networks. IEEE, ??? pp 39–57. https://doi.org/10.1109/SP.2017.49 https://ieeexplore.ieee.org/document/7958570
10. Vemparala M-R, Frickenstein A, Fasfous N, Frickenstein L, Zhao Q, Kuhn S, Ehrhardt D, Wu Y, Unger C, Nagaraja N-S et al (2021) Breakingbed: Breaking binary and efficient deep neural networks by adversarial attacks. In: Proceedings of SAI Intelligent Systems Conference. Springer, pp 148–167
11. Zhu Y, Jiang Y (2021) Imperceptible adversarial attacks against traffic scene recognition. Soft Comput 25(20):13069–13077
12. Deng Y, Zheng X, Zhang T, Chen C, Lou G, Kim M (2020) An analysis of adversarial attacks and defenses on autonomous driving models. In: 2020 IEEE International Conference on Pervasive Computing and Communications (PerCom). IEEE, pp. 1–10. https://doi.org/10.1109/PerCom45495.2020.9127389. https://ieeexplore.ieee.org/document/9127389
13. Zhang F, Chowdhury SP, Christakis M (2020) DeepSearch: a simple and effective blackbox attack for deep neural networks. https://doi.org/10.1145/3368089.3409750
14. Goodfellow IJ, Shlens J, Szegedy C (2014) Published as a conference paper at ICLR 2015 explaining and harnessing adversarial examples.
15. Wu F, Xiao L, Yang W, Zhu J (2020) Defense against adversarial attacks in traffic sign images identification based on 5g. EURASIP J Wireless Commun Netw 2020(1):1–15. https://doi.org/10.1186/s13638-020-01775-5
16. Xu, W., Evans, D., Qi, Y.: Feature squeezing: Detecting adversarial examples in deep neural networks (2017) abs/1704.01155. journal: CoRR; 1704.01155
17. Abdel-Hakim A (2019) Ally patches for spoliation of adversarial patches. J Big Data 6(1):51. https://doi.org/10.1186/s40537-019-0213-4 ID: Abdel-Hakim2019
18. Li Y, Velipasalar S (2020) Weighted average precision: adversarial example detection in the visual perception of autonomous vehicles
19. Review TNL The dangers of driverless cars. https://www.natlawreview.com/article/dangers-driverless-cars. Accessed 05 May 2021
20. Riccio V, Jahangirova G, Stocco A, Humbatova N, Weiss M, Tonella P (2020) Testing machine learning based systems: a systematic mapping. Empirical Softw Eng 25(6):5193–5254
21. Michel A, Jha SK, Ewetz R (2022) A survey on the vulnerability of deep neural networks against adversarial attacks. Prog Artif Intell:1–11
22. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial examples in the physical world (2016) abs/1607.02533. journal: CoRR; 1607.02533
23. Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations (2016) abs/1610.08401. journal: CoRR; 1610.08401

24. Poursaeed, O., Katsman, I., Gao, B., Belongie, S.J.: Generative adversarial perturbations (2017) abs/1712.02328 . journal: CoRR; 1712.02328
25. Robust physical-world attacks on deep learning visual classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018 pp. 1625–1634. https://doi.org/10.1109/CVPR.2018.00175. https://ieeexplore.ieee.org/document/8578273
26. Yan C (2016) Can you trust autonomous vehicles : Contactless attacks against sensors of self-driving vehicle
27. Sitawarin C, Bhagoji AN, Mosenia A, Chiang M, Mittal P (2018) Darts: Deceiving autonomous cars with toxic signs
28. Cao, Y., Xiao, C., Yang, D., Fang, J., Yang, R., Liu, M., Li, B.: Adversarial objects against lidar-based autonomous driving systems (2019) abs/1907.05418. journal: CoRR; 1907.05418
29. Ondruš J, Kolla E, Vertaľ P, Šarić Ž (2020) How do autonomous cars work? Trans Res Proc 44:226–233. https://doi.org/10.1016/j.trpro.2020.02.049 ID: 308315
30. Tian, Y., Pei, K., Jana, S., Ray, B.: Deeptest: Automated testing of deep-neural-network-driven autonomous cars (2017). CoRR abs/1708.08559. 1708.08559
31. Ferreira F, Silva LL, Valente MT (2021) Software engineering meets deep learning: a mapping study. In: Proceedings of the 36th Annual ACM Symposium on Applied Computing, pp 1542–1549
32. Goodfellow I, Bengio Y, Courville A (2016) Deep Learning. MIT Press, ??? http://www.deeplearningbook.org
33. Kiran BR, Sobh I, Talpaert V, Mannion P, Al Sallab AA, Yogamani S, Ṕerez, P. (2021) Deep reinforcement learning for autonomous driving: A survey. IEEE Trans Intell Trans Syst 23(6):4909–4926
34. Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D (2015) Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971
35. Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, Riedmiller M (2013) Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602
36. He X, Yang H, Hu Z, Lv C (2022) Robust lane change decision making for autonomous vehicles: An observation adversarial reinforcement learning approach. In: IEEE Transactions on Intelligent Vehicles
37. He X, Lou B, Yang H, Lv C (2022) Robust decision making for autonomous vehicles at highway on-ramps: A constrained adversarial reinforcement learning approach. In: IEEE Transactions on Intelligent Transportation Systems
38. Behzadan V, Munir A (2019) Adversarial reinforcement learning framework for benchmarking collision avoidance mechanisms in autonomous vehicles. IEEE Intell Trans Syst Mag 13(2):236–241
39. Ma X, Driggs-Campbell K, Kochenderfer MJ (2018) Improved robustness and safety for autonomous vehicle control with adversarial reinforcement learning. In: 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, pp 1665–1671
40. Xu H, Ma Y, Liu H-C, Deb D, Liu H, Tang J-L, Jain AK (2020) Adversarial attacks and defenses in images, graphs and text: a review. Int J Automat Comput 17(2):151–178
41. Shen J, Robertson N (2021) Bbas: Towards large scale effective ensemble adversarial attacks against deep neural network learning. Inform Sci 569:469–478
42. Miller DJ, Xiang Z, Kesidis G (2020) Adversarial learning targeting deep neural network classification: a comprehensive review of defenses against attacks. Proc IEEE 108(3):402–433. https://doi.org/10.1109/JPROC.2020.2970615
43. Hamdi A, Rojas S, Thabet A, Ghanem B (2020) Advpc: Transferable adversarial perturbations on 3d point clouds. In: Vedaldi A, Bischof H, Brox T, Frahm J-M (eds) Computer Vision – ECCV 2020. Springer, Cham, pp 241–257
44. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R (2013) Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199
45. Moosavi-Dezfooli S-M, Fawzi A, Frossard P (2016) Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 2574–2582
46. Papernot N, McDaniel P, Jha S, Fredrikson M, Celik ZB, Swami A (2016) The limitations of deep learning in adversarial settings. In: 2016 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, pp 372–387
47. Carlini N, Wagner D (2017) Towards evaluating the robustness of neural networks. In: 2017 Ieee Symposium on Security and Privacy (sp). IEEE, pp 39–57
48. Athalye A, Engstrom L, Ilyas A, Kwok K (2018) Synthesizing robust adversarial examples. In: International Conference on Machine Learning. PMLR, pp 284–293
49. Guo C, Frank JS, Weinberger KQ (2018) Low frequency adversarial perturbation. arXiv preprint arXiv:1809.08758
50. Dong Y, Liao F, Pang T, Su H, Zhu J, Hu X, Li J (2018) Boosting adversarial attacks with momentum. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 9185–9193
51. Machado GR, Silva E, Goldschmidt RR (2021) Adversarial machine learning in image classification: A survey toward the defender's perspective. ACM Comput Surveys (CSUR) 55(1):1–38
52. Ghiasi, A., Shafahi, A., Goldstein, T.: Breaking certified defenses: semantic adversarial examples with spoofed robustness certificates. (2020) CoRR abs/2003.08937. 2003.08937
53. Su J, Vargas DV, Sakurai K (2019) One pixel attack for fooling deep neural networks. IEEE Trans Evol Comput 23(5):828–841
54. Xie X, Ma L, Juefei-Xu F, Xue M, Chen H, Liu Y, Zhao J, Li B, Yin J, See S (2019) Deephunter: a coverage-guided fuzz testing framework for deep neural networks. In: Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis, pp 146–157
55. Ren H, Huang T, Yan H (2021) Adversarial examples: attacks and defenses in the physical world. Int J Mach Learn Cyber 12(11):3325–3336
56. Hamdi A, Mueller M, Ghanem B (2020) Sada: Semantic adversarial diagnostic attacks for autonomous applications. Proc AAAI Conf Artif Intell 34(7):10901–10908. https://doi.org/10.1609/aaai.v34i07.6722
57. Chen P-Y, Zhang H, Sharma Y, Yi J, Hsieh C-J (2017) Zoo. https://doi.org/10.1145/3128572.3140448
58. Narodytska, N., Kasiviswanathan, S.P.: Simple black-box adversarial perturbations for deep networks (2016) abs/1612.06299 . journal: CoRR; 1612.06299
59. Lecuyer M, Atlidakis V, Geambasu R, Hsu D, Jana S (2019) Certified robustness to adversarial examples with differential privacy. In: 2019 IEEE Symposium on Security and Privacy (SP). IEEE, pp 656–672
60. Zhang H, Chen H, Xiao C, Gowal S, Stanforth R, Li B, Boning D, Hsieh C-J (2019) Towards stable and efficient training of verifiably robust neural networks. arXiv preprint arXiv:1906.06316

61. Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A (2017) Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083
62. Chakraborty A, Alam M, Dey V, Chattopadhyay A, Mukhopadhyay D (2021) A survey on adversarial attacks and defences. CAAI Trans Intell Technol 6(1):25–45
63. Kurakin A, Goodfellow IJ, Bengio S (2018) Adversarial examples in the physical world. In: Artificial Intelligence Safety and Security. Chapman and Hall/CRC, pp. 99–112
64. Khrulkov V, Oseledets I (2018) Art of singular vectors and universal adversarial perturbations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 8562–8570
65. Mopuri KR, Ojha U, Garg U, Babu RV (2018) Nag: Network for adversary generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 742–751
66. Zhang C, Benz P, Lin C, Karjauv A, Wu J, Kweon IS (2021) A survey on universal adversarial attack. arXiv preprint arXiv:2103.01498
67. Zhang C, Benz P, Karjauv A, Kweon IS (2021) Universal adversarial perturbations through the lens of deep steganography: Towards a fourier perspective. Proc AAAI Conf Artif Intell 35:3296–3304
68. Zhang C, Benz P, Karjauv A, Sun G, Kweon IS (2020) Udh: Universal deep hiding for steganography, watermarking, and light field messaging. Adv Neural Inf Process Syst 33:10223–10234
69. Xiao C, Zhu J-Y, Li B, He W, Liu M, Song D (2018) Spatially transformed adversarial examples. arXiv preprint arXiv:1801.02612
70. Laidlaw C, Feizi S (2019) Functional adversarial attacks. Adv Neural Inf Process Syst 32
71. Pei K, Cao Y, Yang J, Jana S (2017) Deepxplore: Automated whitebox testing of deep learning systems. In: Proceedings of the 26th Symposium on Operating Systems Principles, pp 1–18
72. Zantedeschi, V., Nicolae, M.-I., Rawat, A.: Efficient defenses against adversarial attacks (2017) abs/1707.06728 . journal: CoRR; 1707.06728
73. Guo, C., Rana, M., M Cisse, van der Maaten, L.: Countering adversarial images using input transformations (2017) abs/1711.00117. journal: CoRR; 1711.00117
74. Bhardwaj K, Gope D, Ward J, Whatmough P, Loh D (2022) Super-efficient super resolution for fast adversarial defense at the edge. In: 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, pp 418–423
75. Athalye A, Carlini N, Wagner D (2018) Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples
76. Kwon H, Lee J (2021) Diversity adversarial training against adversarial attack on deep neural networks. Symmetry 13(3):428
77. Xie C, Wang J, Zhang Z, Ren Z, Yuille A (2017) Mitigating adversarial effects through randomization. arXiv preprint arXiv:1711.01991
78. Sutanto RE, Lee S (2020) Adversarial attack defense based on the deep image prior network. In: Kim KJ, Kim H-Y (eds) Information Science and Applications. Springer, Singapore, pp 519–526
79. Hosseini H, Kannan S, Poovendran R (2019) Dropping pixels for adversarial robustness. IEEE, pp. 91–9. https://doi.org/10.1109/CVPRW.2019.00017. https://ieeexplore.ieee.org/document/9025677
80. Carlini N, Wagner D (2017) Adversarial examples are not easily detected. AISec 39;17. ACM, pp. 3–14. https://doi.org/10.1145/3128572.3140444. http://dl.acm.org/citation.cfm?id61;3140444
81. Sun Q, Rao AA, Yao X, Yu B, Hu S (2020) Counteracting adversarial attacks in autonomous driving. In: 2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD), pp 1–7 ID: 1
82. Liao F, Liang M, Dong Y, Pang T, Hu X, Zhu J (2018) Defense against adversarial attacks using high-level representation guided denoiser. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1778–1787
83. Khamis, R.A., Shafiq, M.O., Matrawy, A.: Investigating resistance of deep learning-based IDS against adversaries using min-max optimization (2019). CoRR abs/1910.14107 . 1910.14107
84. Raghunathan A, Steinhardt J, Liang P (2018) Published as a conference paper at ICLR 2018 certified defenses against adversarial examples.
85. Hashemi AS, Mozaffari S (2021) Cnn adversarial attack mitigation using perturbed samples training. Multimed Tools Appl 80(14):22077–22095
86. Hu S, Nalisnick E, Welling M (2022) Adversarial defense via image denoising with chaotic encryption. arXiv preprint arXiv:2203.10290
87. Guo C, Rana M, Cisse M, Van Der Maaten L (2017) Countering adversarial images using input transformations. arXiv preprint arXiv:1711.00117
88. Zhang Y, Li H, Zheng Y, Yao S, Jiang J (2021) Enhanced dnns for malware classification with gan-based adversarial training. J Comput Virol Hack Tech 17(2):153–163
89. Samangouei P, Kabkab M, Chellappa R (2018) Defense-gan: Protecting classifiers against adversarial attacks using generative models. arXiv preprint arXiv:1805.06605
90. Shen S, Jin G, Gao K, Zhang Y (2017) Ape-gan: Adversarial perturbation elimination with gan. arXiv preprint arXiv:1707.05474
91. Liu X, Cheng M, Zhang H, Hsieh C-J (2018) Towards robust neural networks via random self-ensemble. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 369–385
92. Liu W, Shi M, Furon T, Li L (2020) Defending adversarial examples via dnn bottleneck reinforcement. In: Proceedings of the 28th ACM International Conference on Multimedia, pp 1930–1938
93. Cho S, Jun TJ, Oh B, Kim D (2020) Dapas: denoising autoencoder to prevent adversarial attack in semantic segmentation. In: 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, pp 1–8
94. Gu S, Rigazio L (2014) Towards deep neural network architectures robust to adversarial examples. arXiv preprint arXiv:1412.5068
95. Hendrycks D, Gimpel K (2016) Early methods for detecting adversarial images. arXiv preprint arXiv:1608.00530

96. Li X, Li F (2017) Adversarial examples detection in deep networks with convolutional filter statistics. In: Proceedings of the IEEE International Conference on Computer Vision, pp 5764–5772
97. Liu Z, Liu Q, Liu T, Xu N, Lin X, Wang Y, Wen W (2019) Feature distillation: Dnn-oriented jpeg compression against adversarial examples. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 860–868
98. Niu Z, Chen Z, Li L, Yang Y, Li B, Yi J (2020) On the limitations of denoising strategies as adversarial defenses. arXiv preprint arXiv:2012.09384
99. Qiu H, Zeng Y, Zheng Q, Guo S, Zhang T, Li H (2021) An efficient preprocessing-based approach to mitigate advanced adversarial attacks. In: IEEE Transactions on Computers
100. Das N, Shanbhogue M, Chen S-T, Hohman F, Li S, Chen L, Kounavis ME, Chau DH (2018) Shield: Fast, practical defense and vaccination for deep learning using jpeg compression. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 196–204
101. Xu W, Evans D, Qi Y (2017) Feature squeezing: detecting adversarial examples in deep neural networks. arXiv preprint arXiv:1704.01155
102. Prakash A, Moran N, Garber S, DiLillo A, Storer J (2018) Deflecting adversarial attacks with pixel deflection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 8571–8580
103. Xiao C, Deng R, Li B, Lee T, Edwards B, Yi J, Song D, Liu M, Molloy I (2019) Advit: Adversarial frames identifier based on temporal consistency in videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 3968–3977
104. Du X, Xie X, Li Y, Ma L, Liu Y, Zhao J (2019) Deepstellar: Model-based quantitative analysis of stateful deep learning systems. In: Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp 477–487
105. Ribeiro, M.T., Wu, T., Guestrin, C., Singh, S.: Beyond accuracy: Behavioral testing of NLP models with checklist (2020). CoRR abs/2005.04118. 2005.04118
106. Guo, Q., Chen, S., Xie, X., Ma, L., Hu, Q., Liu, H., Liu, Y., Zhao, J., Li, X.: An empirical study towards characterizing deep learning development and deployment across different frameworks and platforms (2019). CoRR abs/1909.06727. 1909.06727
107. Balakrishnan A, Puranic AG, Qin X, Dokhanchi A, Deshmukh JV, Amor HB, Fainekos G (2019) Specifying and evaluating quality metrics for vision-based perception systems. In: 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, pp 1433–1438
108. Meyer D (2019) Uber Cleared Over Arizona Pedestrian's Self-Driving Car Death. http://fortune.com/2019/03/06/uber-clearedarizona-self-driving-death/. Accessed 6 Mar 2019.
109. Shamsabadi AS, Oh C, Cavallaro A (2020) Edgefool: an adversarial image enhancement filter. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp 1898–1902
110. Zhang H, Yu Y, Jiao J, Xing E, Ghaoui LE, Jordan M (2019) Theoretically principled trade-off between robustness and accuracy. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research. PMLR, vol. 97, pp. 7472–7482. http://proceedings.mlr.press/v97/zhang19p.html
111. Wu K, Yu Y (2019) Understanding adversarial robustness: The trade-off between minimum and average margin
112. Weng L, Chen P-Y, Nguyen L, Squillante M, Boopathy A, Oseledets I, Daniel L (2019) Proven: Verifying robustness of neural networks with a probabilistic approach. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 6727–6736. http://proceedings.mlr.press/v97/weng19a.html
113. Xiang C, Qi CR, Li B (2019) Generating 3d adversarial point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 9136–9144
114. Zhou H, Chen K, Zhang W, Fang H, Zhou W, Yu N (2019) Dup-net: Denoiser and upsampler network for 3d adversarial point clouds defense. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 1961–1970
115. Tsai T, Yang K, Ho T-Y, Jin Y (2020) Robust adversarial objects against deep learning models. Proc AAAI Conf Artif Intell 34:954–962

Publisher's Note