1 Security of AI Systems

Liang Tong, Stellar Cyber
Yevgeniy Vorobeychik, Washington University in Saint Louis

In previous chapters, we covered the general privacy and security concerns associated with AI systems and the game-theoretic perspective on these issues. Here, we discuss AI system (particularly machine learning system) vulnerabilities from a security perspective, emphasizing *integrity*. Integrity is a necessary criterion for trustworthy AI—an AI system that demonstrates integrity operates correctly as defined by its designer and can therefore be trusted. Our objective is to understand how AI systems can fail in adversarial environments where an attacker seeks to compromise its integrity. An important tool in achieving this objective is a *threat model*. For a machine learning (ML) system, a threat model is a structured and systematic approach for identifying potential threats and vulnerabilities that may compromise its integrity. It involves a comprehensive analysis and assessment of potential risks and the adversaries that may exploit them. Specifically, the construction of a threat model typically involves several key aspects.

- The *adversary's knowledge* about the targeted ML system, such as its access to training data and machine learning model.
- The *adversary's capabilities* such as its budget for executing attacks.
- The *adversary's goal* to achieve, such as avoiding being detected or misclassified as a targeted label.

Once a threat model has been constructed, it can be utilized to identify ML system vulnerabilities and countermeasures.

In this chapter, we limit the scope of our consideration of vulnerability to *decision-time attacks*, in which an adversary manipulates the input to a deployed ML system to affect its prediction. We then use two domains as concrete case studies: *PDF malware detection* and *face recognition*.

1.1 Decision-Time Attacks

In the (supervised) machine learning literature, it is common to abstract the problem as follows. We are given a training dataset $\mathsf{D} = \{(\boldsymbol{x}_i, y_i)\}$, where $\boldsymbol{x}_i \in \mathsf{X} \subseteq \mathbb{R}^n$ are numeric feature vectors in some feature space X, and $y_i \in \mathsf{L}$ are labels in a label space L. Each data point (or example) in the dataset D is assumed to be generated in an independent and identically distributed (i.i.d.) manner according to an unknown distribution P. In addition, we are given a hypothesis (model) space, H. Our objective is to identify (*learn*) a good model $h_{\boldsymbol{\theta}} \in \mathsf{H}$ parameterized by $\boldsymbol{\theta}$ in the sense that it yields a minimal error on new examples drawn from P. Since P is unknown in practice, one typically aims to find $h_{\boldsymbol{\theta}} \in \mathsf{H}$ which (approximately) minimizes empirical error on the training data D.

To undermine the integrity of an ML model, a decision-time attack is usually employed by an adversary to induce predicting mistakes after the model has been deployed. To achieve this goal, the adversary manipulates an adversarial example by transforming $\boldsymbol{x}$ to $\boldsymbol{x}'$ such that $\boldsymbol{x}'$ is similar to $\boldsymbol{x}$ but the model makes an incorrect prediction on $\boldsymbol{x}'$ (while making a correct prediction on the original input $\boldsymbol{x}$). A key distinction we make in this chapter is between two types of threat models. The first type, *feature-space attacks*, modify features directly. The second, *realizable attacks*, account for practical domain constraints that limit what kinds of attacks can be executed, and impact attack efficacy.

1.1.1 Feature-Space Attacks

As the name implies, feature-space attacks work directly in the *feature space*, as is typical in the machine learning literature [Athalye et al. 2018, Biggio et al. 2013, Carlini and Wagner 2017, Dalvi et al. 2004, Goodfellow et al. 2015, Lowd and Meek 2005, Zhang et al. 2015], including the application of machine learning (especially, deep neural networks) in computer vision tasks [Carlini and Wagner 2017, Goodfellow et al. 2015, Moosavi-Dezfooli et al. 2016, Papernot et al. 2016, Szegedy et al. 2014].

In feature-space attacks, the attacker is *modeled* as starting with a malicious feature vector $\boldsymbol{x}$ and *directly modifying the features* to produce another feature vector $\boldsymbol{x}' \in \mathsf{X}$, to yield erroneous predictions, i.e., $y' = h_{\boldsymbol{\theta}}(\boldsymbol{x}')$ (for example, being mislabeled as benign). Crucially, because feature vectors are an abstract representation of the underlying problem instances (such as scenes being photographed in computer vision, or files in the context of malware detection), we must abstract the notion of preserving (malicious) functionality. This is accomplished through the use of a cost function, $c(\boldsymbol{x},\boldsymbol{x}')$, whereby the attacker is penalized for greater modifications to the given feature vector $\boldsymbol{x}$, typically measured using an ℓ_p norm difference between the original malicious instance and the modified feature vector [Biggio et al. 2013, Li and Vorobeychik 2018]. Within computer vision, a common motivation of the use of ℓ_p norm is that when the norm of the adversarial perturbation is small, it is imperceptible to a human. We term these the *feature-space attacks*.

The construction of an adversarial example in feature space for $\boldsymbol{x}$ can be represented by the following optimization problem (or its variants):

$$\max_{\boldsymbol{\delta} \in \Delta(\epsilon)} \mathcal{L}\left(h_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), y\right), \tag{1.1}$$

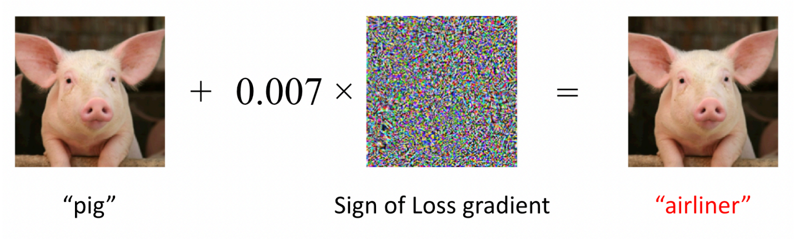

Figure 1.1 An illustration of the FGSM attack [Goodfellow et al. 2015]. The original image is correctly classified as a pig. After adding the imperceptible FGSM perturbation, the image is misclassified as an airliner. The perturbation is magnified for better visualization.

where $\boldsymbol{\delta}$ is the adversarial perturbations to be added on $\boldsymbol{x}$. $\mathcal{L}(\cdot)$ is the loss function used to train the classifier $h_{\boldsymbol{\theta}}$. $\Delta(\varepsilon)$ is the feasible perturbation space which is commonly represented as a ℓ_p ball: $\Delta(\varepsilon) = \{\boldsymbol{\delta} : \|\boldsymbol{\delta}\|_p \leq \varepsilon\}$. Many solutions have been proposed to solve the aforementioned optimization problem. Next, we present several prominent examples.

FGSM. The *Fast Gradient Sign Method (FGSM)* [Goodfellow et al. 2015] is one of the earliest paradigms for constructing adversarial examples in the feature space, illustrated in Figure 1.1. FGSM assumes that constraints on adversarial perturbations $\boldsymbol{\delta}$ are imposed in terms of the ℓ_∞ norm, i.e., $\Delta(\varepsilon) = \{\boldsymbol{\delta} : \|\boldsymbol{\delta}\|_\infty \leq \varepsilon\}$, and uses a linear approximation to compute the perturbations in closed form as

$$\boldsymbol{\delta}^* = \varepsilon \cdot \text{sign}(\nabla_{\boldsymbol{\delta}} \mathcal{L}(h_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), y)). \tag{1.2}$$

While FGSM was shown to be quite effective initially, it ultimately proved to be a relatively weak approach for generating adversarial perturbations. Next, we describe three considerably stronger approaches: PGD, DeepFool, and CW.

PGD. The key limitation of FGSM is that it effectively takes only a single gradient step. A natural generalization is to perform FGSM iteratively, each time projecting the gradient step to the feasible region (in terms of both the feature space and magnitude of the perturbation. This approach has come to be known simply as *Projected Gradient Descent (PGD)* [Madry et al. 2018] (even though it typically performs gradient ascent, as in the case of the objective represented in Problem (1.1). Specifically, each PGD iteration performs the following gradient update, followed by a projection step:

$$\boldsymbol{\delta}_{t+1} = \boldsymbol{\delta}_t + \alpha \cdot \varepsilon \cdot \text{sign}(\nabla_{\boldsymbol{\delta}_t} \mathcal{L}(h_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}_t), y)), \tag{1.3}$$

where α is the step size. We can initialize $\boldsymbol{\delta}_0 = \mathbf{0}$, although a more effective variant also considers random starts with $\boldsymbol{\delta}_0$ initialized to a small random vector. PGD has proved to be one of the most powerful attacks in practice, particularly if we use a sufficient number

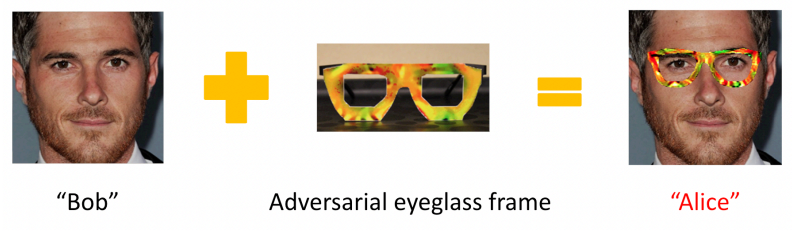

Figure 1.2 An illustration of realizable attacks on face recognition [Sharif et al. 2016b]. A person's face image is misidentified as that of another when he employs an adversarial eyeglass frame.

of iterations and random starts, and remains one of the main ways to empirically evaluate robustness of models to adversarial perturbations with ℓ_p-norm constraints.

CW. Another powerful class of attacks was designed by Carlini and Wagner [2017]; we refer to these simply as CW. The most basic variant of CW attacks assumes that adversarial perturbation constraints are in terms of the ℓ_2-norm bounds. CW formulates the optimization problem for identifying adversarial perturbations as follows:

$$\min ||\boldsymbol{\delta}||_2 + c \cdot f(\boldsymbol{x} + \boldsymbol{\delta}), \tag{1.4}$$

where t is the target label of prediction. f is a function such that $h_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}) = t$ only if $f(\boldsymbol{x} + \boldsymbol{\delta}) \leq 0$. There are many possible choices for f, with the one that yields the best performance being

$$f(\boldsymbol{x}') = \max(\max_{i \neq t} Z(\boldsymbol{x}')_i - Z(\boldsymbol{x}')_t, 0), \tag{1.5}$$

where Z returns the logits of the last layer of $h_{\boldsymbol{\theta}}$. Then, the optimization problem in Eq.(1.4) can be solved by any gradient-based method.

Coordinate Greedy. When the features are binary, the optimization problem in Eq. (1.1) can be solved by using *Coordinate Greedy* (also known as an iterative improvement) [Li and Vorobeychik 2018], which optimizes one randomly chosen coordinate of the feature vector at a time, until a local optimum is reached. In order to improve the quality of the resulting solution (given that $\mathcal{L}(\cdot)$ is typically non-concave), the above process can be repeated from multiple random starting points [Li and Vorobeychik 2018, Madry et al. 2018], just as described above for PGD attacks.

1.1.2 Realizable Attacks

Feature-space attacks are clearly an abstraction. But an abstraction of what? In reality, the machine learning pipeline does not begin with features—it begins with *entities*. In the context of malware detection, entities are typically executable files. One prepares datasets for training malware detectors by first defining feature extraction functions, which then map each file to an associated feature vector. In the context of computer vision, on the other hand, entities are scenes, which are translated into features when a camera photographs the scene. We can formalize this by letting e denote the actual entity, with $\boldsymbol{x}(e)$ the feature vector extracted from it. A *realizable attack* is an attack directly on the entity e, which indirectly (through a feature extractor, such as a camera taking the photograph) impacts its feature vector $\boldsymbol{x}$. Realizable attacks must abide by practical constraints imposed by the fact that an actual entity is modified, rather then feature values. For example, when a scene is modified, it is not meaningful to talk about perturbations to pixels; rather, modifications are, for example, to objects in the scene.

In the context of cybersecurity, an early realizable attack on machine learning was devised by Fogla et al. [2006] who attack an anomaly-based intrusion detection systems. Šrndic and Laskov [2014] present a case study of an evasion attack against PDFRate [Smutz and Stavrou 2012], a state-of-the-art PDF malware classifier, while [Xu et al. 2016] propose EvadeML, a fully realizable attack on PDF malware classifiers that generates evasion instances by directly modifying PDF source using genetic programming and a sandbox to ensure malicious functionality.

In computer vision, two noteworthy examples of realizable (or physical) attacks involve adversarial occlusions placed on objects in the scene. Sharif et al. [2016a] design printable eyeglass frames with adversarial noise inside them, as illustated in Figure 1.2, with the goal of face images being misclassified as a different person than actually photographed. Eykholt et al. [2018], on the other hand, design adversarial stickers that can be printed and placed on road signs (notably, a stop sign) with the goal of these being misclassified (e.g., a stop sign incorrectly classified as a speed limit sign).

Next, we discuss the design of realizable attacks on face recognition to illustrate these concepts.

1.2 Illustration: Face Recognition

1.2.1 Face Recognition Systems

In computer vision tasks such as face recognition, the raw feature extractor can be a camera that captures entities and translate these into images with pixels in the digital space. The images are subsequently fed into deep neural networks to extract higher-level features such as shapes and edges. Generally, deep face recognition systems aim to solve the following two tasks: 1) *Face identification*, which returns the predicted identity of a test face image, and 2) *Face verification*, which indicates whether a test (also called probe) face image and the face image stored in the gallery belong to the same person. Based on whether all test identities are predefined in the training set, face recognition systems can be further categorized into *closed-set systems* and *open-set systems* [Liu et al. 2017], as illustrated in Fig. 1.3.

In closed-set face recognition tasks, all the test samples' identities are enrolled in the training set. Specifically, a face identification task is equivalent to a *multi-class classification* problem [Sun et al. 2014a,b, Taigman et al. 2014]. A face verification task is a natural

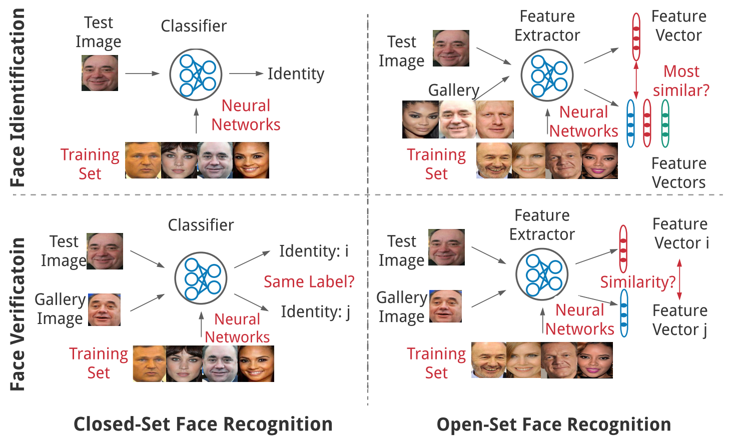

Figure 1.3 Closed-set and open-set face recognition systems.

extension of face identification by first performing the classification twice (one for the test image and the other for the gallery) and then comparing the predicted identities.

In contrast, there are usually no overlaps among identities in the training and test set for open-set tasks. In this setting, a face verification task is essentially a *metric learning* problem, which aims to maximize *intra-class distance* and minimize *inter-class distance* under a chosen metric space in two steps [Deng et al. 2019, Liu et al. 2016, 2017, Parkhi et al. 2015, Schroff et al. 2015, Wen et al. 2016]. First, we train a feature extractor that maps a face image into a discriminative feature space by using a carefully designed loss function. Second, we measure the distance between feature vectors of the test and gallery face images to see if it is above a verification threshold. As an extension of face verification, the face identification task requires additional steps to compare the distances between the feature vectors of the test image and each gallery image, and then choose the gallery's identity corresponding to the shortest distance. Here, we limit our scope to face identification for closed-set systems, as threat models for face verification tasks and open-set systems are just extensions of these.

1.2.2 Threat Models for Face Recognition

We focus on vulnerability of face recognition to both feature-space attacks and *the digital representation of physically realizable attacks* (henceforth, we'll refer to them simply as *physically realizable attacks*). Specifically, physically realizable attacks are digital attacks that can produce adversarial perturbations with low suspiciousness, and these perturbations can be realized in the physical world by using techniques such as 3-D printing (*e.g.*, Fig. 1.4 illustrates one example of such attacks on face recognition systems). Compared to realizable attacks ac-

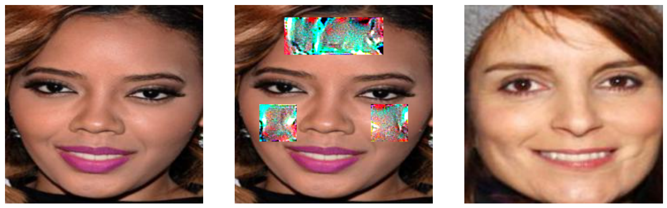

Figure 1.4 Sticker attack: an example of physically realizable attacks on face recognition systems. Left: original input image. Middle: adversarial sticker on the face. Right: predicted identity. In practice, the adversarial stickers can be printed and put on human faces.

tually implemented in the scene, physically realizable attacks can evaluate robustness of face recognition systems more efficiently. On the one hand, such attacks allow us to iteratively modify digital images directly so the evaluation can be significantly sped up compared to modifying real-world objects and then photographing them. On the other hand, robustness to physically realizable attacks provides a lower bound on robustness to analogous attacks in the physical world, as the former has fewer constraints and larger solution space.

Formally, both feature-space and physically realizable attacks can be performed by solving the following general form of an optimization problem (*e.g.*, for closed-set identification task):

$$\arg\max_{\boldsymbol{\delta}} \mathcal{L}(S(\boldsymbol{x}+M\boldsymbol{\delta}), y) \qquad s.t.\ \boldsymbol{\delta} \in \Delta, \tag{1.6}$$

where S is the target face recognition model, $\mathcal{L}$ is the adversary's utility function (*e.g.*, the loss function used to train S), $\boldsymbol{x}$ is the original input face image, y is the associated identity, $\boldsymbol{\delta}$ is the adversarial perturbation, and Δ is the feasible space of the perturbation. Here, M denotes the mask matrix that constrains the area of perturbation; it has the same dimension as $\boldsymbol{\delta}$ and contains 1s where the perturbation is allowed and 0s where there is no perturbation.

Next, we present several threat models for face recognition systems, categorized along the following dimensions:

- *Perturbation type*, such as perturbations produced by pixel-level feature-space attacks and physically realizable attacks.
- *The adversary's knowledge of the target system*, *i.e.*, the information about which subcomponents of S are leaked to the adversary.
- *The adversary's goal*, such as the circumvention of detection (dodging) and the misrecognition as a target identity (impersonation).
- *The adversary's capability*. For example, an attacker can either individually perturb each input face image, or produce universal perturbations for images batch-wise.

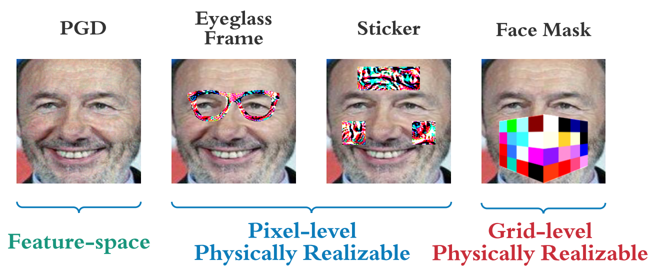
PGD
Eyeglass Frame
Sticker
Face Mask
Feature-space
Pixel-level Physically Realizable
Grid-level Physically Realizable

Figure 1.5 Threat models by perturbation types.

1.2.2.1 Perturbation Type

In general, there are three categories of threat models using different perturbation types: *digital attack*, *pixel-level physically realizable attack*, and *grid-level physically realizable attack*, as shown in Figure 1.5.

- *Feature-Space Attacks*. Feature-space attacks produce small perturbations on the entire input face image, such as the PGD attack [Madry et al. 2018] that uses the ℓ_∞-norm, and the CW attack [Carlini and Wagner 2017] which is an ℓ_2 attack, described in Section 1.1.1.
- *Pixel-level Physically Realizable Attacks*. This category of attacks features pixel-level perturbations that can be realized in the physical world (*e.g.*, by printing them on glossy photo papers). In this case, the attacker adds large pixel-level perturbations to a small area of the input image (*e.g.*, face accessories such as an eyeglass frame or face stickers). Such attacks have been demonstrated to successfully fool VGG-based face recognition systems [Sharif et al. 2016b].
- *Grid-level Physically Realizable Attacks*. In practice, pixel-level perturbations are not printable on face accessories made of *coarse* materials, such as face masks using cloths and non-woven fabrics. To address this issue, grid-level physically realizable face mask attacks add a color grid on face masks, as shown in Figure 1.5.

1.2.2.2 Adversary's System Knowledge

The key components of a face recognition system S are the training set D and neural architecture h. It is natural to categorize threat models based on the adversary's knowledge of these two components. From the attackers' perspective, threat models can be classified into the following scenarios:

- *Zero Knowledge*. Both D and h are invisible to the adversary. This is the weakest adversarial setting, as no critical information of S is leaked. Thus, it provides a lower

Table 1.1 Attack success rate of dodging attacks on closed-set face recognition systems by the attacker's system knowledge [Tong et al. 2021]. Z represents zero knowledge, T is training set, A is neural architecture, and F represents full knowledge.

Target System	**Attack Type**	**Attacker's System Knowledge**			
		Z	**T**	**A**	**F**
VGGFace [Parkhi et al. 2015]	PGD	0.40	0.51	0.93	0.94
	Eyeglass Frame	0.23	0.28	0.70	0.99
	Sticker	0.05	0.06	0.47	0.98
	Face Mask	0.26	0.32	0.63	1.00
FaceNet [Schroff et al. 2015]	PGD	0.83	0.83	1.00	1.00
	Eyeglass Frame	0.13	0.16	0.90	1.00
	Sticker	0.01	0.01	0.92	1.00
	Face Mask	0.30	0.42	0.83	1.00
ArcFace18 [Deng et al. 2019]	PGD	0.87	0.92	0.97	1.00
	Eyeglass Frame	0.06	0.06	0.44	1.00
	Sticker	0.01	0.01	0.37	1.00
	Face Mask	0.27	0.33	0.71	1.00
ArcFace50 [Deng et al. 2019]	PGD	0.87	0.90	0.81	0.99
	Eyeglass Frame	0.09	0.12	0.44	0.99
	Sticker	0.00	0.01	0.14	0.94
	Face Mask	0.29	0.36	0.67	0.99
ArcFace101 [Deng et al. 2019]	PGD	0.81	0.78	0.86	0.96
	Eyeglass Frame	0.03	0.03	0.26	0.98
	Sticker	0.04	0.04	0.08	0.95
	Face Mask	0.26	0.36	0.54	0.99

bound for the vulnerability of S. In this scenario, the attacks are referred to as *black-box attacks*, where the adversary needs no internal details of S to compromise it.

- *Training Set*. This scenario enables the assessment of the robustness of the training set of S in adversarial settings. Here, only the training set D is visible to the adversary. Without knowing h, an adversary constructs a surrogate system S' by training a surrogate neural architecture h' on D, *i.e.,* $S' = f(h'; D)$. Then, the adversary performs the aforementioned transfer-based attack on S' and evaluates S using the transferred adversarial examples.
- *Neural Architecture*. The adversary only knows the neural architecture h of S but has no access to the training set D., This enables the evaluation of the vulnerability of the neural architecture h of S. Without knowing D, the attacker can build its surrogate system $S' = f(h; D')$ and conduct the transfer-based attack to evaluate S.
- *Full knowledge*. In the worst case, the adversary can have an accurate knowledge of both the training set D and neural architecture h. Thus, it provides an upper bound for

vulnerability evaluation on *S*. In this scenario, the attacker can fully reproduce *S* offline and then perform *white-box attacks* on *S*.

Recently, we investigated and quantified the vulnerability of different components of open-source face recognition systems in the face of decision-time attacks under different perturbation types and adversary's knowledge about them [Tong et al. 2021]; the data is summarized in Table 1.1. We can observe that *the neural architecture is significantly more fragile than the training set in most adversarial settings*. For example, when only the neural architecture is exposed to the attacker, the sticker attack has a high success rate of 0.92 on FaceNet. In contrast, when the attacker only knows the training set, the attack success rate drops to 0.01. In addition, by comparing each row of Table 1.1 that corresponds to the same target system, one can observe that *feature-space attacks (PGD) are considerably more potent than their physically realizable counterparts on closed-set systems, while grid-level perturbations on face masks are noticeably more effective than pixel-level physically realizable perturbations (i.e., the eyeglass frame attack and the sticker attack).*

1.2.2.3 Adversary's Goal

In addition to the adversary's knowledge about the target face recognition system, threat models can differ in specific goals detailed below.

- *Dodging (Untargeted)*. In a dodging attack, an adversary aims to have his/her face misidentified as another arbitrary face. For example, the adversary can be a terrorist who wants to bypass a face recognition system for biometric security checking. As the dodging attack has no specific target identity, it is also called the *untargeted attack*.
- *Impersonation (Targeted)*. In an impersonation, or targeted, attack, an adversary seeks to produce an adversarial example that cases the face to be incorrectly recognized as a target identity. For example, the adversary may try to camouflage his/her face to be identified as an authorized user of a laptop that uses face recognition for authentication.

We observed that *while both attacks are highly effective in compromising different face recognition systems (with nearly* 100% *attack success rate), dodging attacks exhibit stronger transferability in black-box attacks than impersonation attacks* [Tong et al. 2021].

1.2.2.4 Adversary's Capability

In practice, even when the adversaries share the same system knowledge and goal, their capabilities can still differ due to time and/or budget constraints, such as the budget for printing adversarial eyeglass frames [Sharif et al. 2016b]. We can consider two categories of attacker capabilities: *individual attack* and *universal attack*.

- *Individual attack*. The adversary has sufficient time and budget to produce a specific perturbation for each input face image.

- *Universal attack*. The adversary has a time/budget constraint such that he/she can only generate a *universal (face-agnostic)* perturbation that fools a face recognition system for a batch of face images rather than for every input.

In a universal attack, the adversary can explicitly control the universality of the perturbation by setting different values of N, the number of face images that share the same physically realizable perturbation. In general, a larger N corresponds to a lower budget because more images share the perturbation, thereby imposing more constraints on the attack and decreasing its success rate. We showed that *most face recognition systems are susceptible to universal attacks even when N is as large as 20* [Tong et al. 2021],. Under these conditions, a universal attack can still obtain a success rate above 50%.

1.3 Robust Learning: Approaches and Efficacy

There have been numerous defense approaches to make ML robust. Here, we present two classes of methods to learn robust models that have proved both sufficiently scalable and effective even against adaptive attacks: *adversarial training* [Cohen et al. 2019, Goodfellow et al. 2015, Madry et al. 2018, Szegedy et al. 2014] and *randomized smoothing* [Cohen et al. 2019, Lecuyer et al. 2019].

Adversarial Training. The fundamental idea of adversarial training is to solve the following robust optimization problem:

$$\min_{\boldsymbol{\theta}} \frac{1}{|D|} \sum_{\boldsymbol{x},y \in D} \max_{\|\boldsymbol{\delta}\|_p \leq \varepsilon} \mathcal{L}\left(h_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), y\right), \tag{1.7}$$

where D is the training dataset. In practice, this problem is commonly solved by iteratively using the following two steps: 1) compute or approximate the solution to the inner maximization problem, and 2) use this solution to update the model parameters $\boldsymbol{\theta}$.

In most settings, the inner optimization problem is non-convex, and quite challenging to solve to optimality. There are two training paradigms that address this challenge. The first is to compute heuristic solutions as lower bounds to

$$\max_{\|\boldsymbol{\delta}\|_p \leq \varepsilon} \mathcal{L}\left(h_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), y\right). \tag{1.8}$$

Commonly, one would simply use one of the methods for generating adversarial perturbations—typically, feature-space attacks. For example, PGD has been shown to be an effective technique for generating heuristic solutions. More recently, a variant of FGSM in which we initialize $\boldsymbol{\delta}$ to a random start, rather than 0, has been shown to be highly effective as well, and considerably less expensive than PGD [Wong et al. 2020].

An alternative paradigm is to make use of provably upper bounds on the solution to Problem (1.8). These are often intimately tied to the goal of *certifying*, or proving, robustness of the model at decision time. For example, if the model $h_{\boldsymbol{\theta}}(\boldsymbol{x})$ is a neural network with ReLU activations, the problem of verifying robustness can be formulated as a mixed-integer linear program, and one can further make use of convex relaxations of the ReLU neurons to obtain a linear program. Moreover, Wong and Kolter [2018] show that one can make use of a dual of this linear program, along with weak duality, to obtain a differentiable upper bound on the inner optimization problem. This upper bound can thus be used directly as a part of the stochastic gradient descent procedure of learning the model parameters.

Randomized Smoothing. The second class of methods for robust learning considers adding random perturbations to inputs at both training and test time. The basic idea is to construct a new smoothed classifier $g_{\boldsymbol{\theta}}(\cdot)$ from a base classifier $h_{\boldsymbol{\theta}}(\cdot)$ as follows: first, the base classifier $h_{\boldsymbol{\theta}}(\cdot)$ is trained with *Gaussian data augmentation* with variance $\boldsymbol{\sigma}^2$; then, for any input $\boldsymbol{x}$ at test time, the smoothed classifier $g_{\boldsymbol{\theta}}(\cdot)$ returns the class that has the highest probability measure for the base classifier $h_{\boldsymbol{\theta}}(\cdot)$ when inputs are perturbed with isotropic Gaussian noise:

$$\begin{aligned} g_{\boldsymbol{\theta}}(\boldsymbol{x}) &= \arg\max_{c} P(h_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\eta}) = c) \\ \text{where } \boldsymbol{\eta} &\sim N\left(\mathbf{0}, \sigma^2 \boldsymbol{I}\right) \end{aligned} . \tag{1.9}$$

Indeed, the approaches for robust machine learning have generally proved quite effective in boosting both empirical and provable robustness under the ℓ_p-norm attacks [Cohen et al. 2019, Madry et al. 2018, Wong and Kolter 2018]. However, their efficacy is less evident in the context of robustness to realizable attacks.

In particular, we have investigated robustness of adversarial training that makes use of PGD to approximate the inner optimization problem (1.8) in yielding robustness to realizable attacks on PDF malware detection [Tong et al. 2019], demonstrating that their efficacy out of the box can vary. Moreover, our exploration of this issue in the context of image classification has shown such approaches, as well as randomized smoothing techniques, to yield poor robustness to attacks involving physical occlusions [Wu et al. 2020]. Both Tong et al. [2019] and Wu et al. [2020] propose approaches that make use of the general adversarial training paradigm, but with modified adversarial models (i.e., versions of the inner problem), which significantly boost the ability to learn classifiers that are robust to realizable attacks.

Additionally, we recently explored this issue in the context of face recognition [Tong et al. 2021], further investigating the *defense against occlusion attacks (DOA)* approach proposed by Wu et al. [2020]. Figure 1.6 presents the vulnerability of DOA-trained face recognition systems with different neural architectures in the face of different realizable attacks. It can be seen that *DOA fails to defend against the grid-level perturbations on face masks for most neural architectures*. Specifically, face mask attacks can achieve > 0.7 success rates on four out of the five face recognition systems refined by DOA. Moreover, we find that *adversarial robustness against one type of perturbation can not be generalized to other types*. For example, while VGGface-DOA exhibits a relatively high level of robustness (more than a 70% accuracy) against pixel-level perturbations (*i.e.*, stickers and eyeglass frames), it is

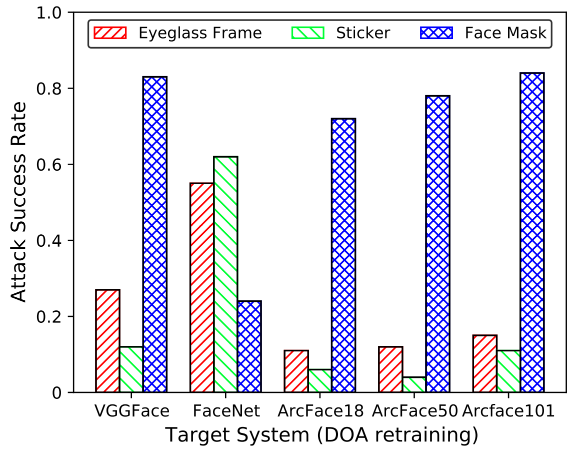

Figure 1.6 Attack success rate of dodging physically realizable attacks on closed-set systems with DOA retraining.

very vulnerable to grid-level perturbations (*i.e.*, face masks). In contrast, using DOA on FaceNet can successfully defend against face mask perturbations, with the attack success rate significantly dropping from 1.0 to 0.24, but it's considerably less effective against eyeglass frames and stickers. Thus, we generally find that it is extremely difficult to obtain a "silver bullet" that can be universally effective, independent of domain, neural architecture, or the nature of the attack.

1.4 Summary

This chapter focuses on AI/ML system vulnerabilities from a security perspective. Starting with general threat models, we introduced two representative attacks that can effectively compromise the integrity of ML systems: the feature-space attack and the realizable attack. We then elucidate these threat models in the context of two domains: PDF malware detection and face recognition, both of which have been shown to be vulnerable to a variety of attacks and for which numerous paradigms for robust ML systems have been proposed. The robustness of these robust ML systems is then discussed. We demonstrate that these systems are still susceptible to a variety of realizable attacks, and that robustness against one type of attack cannot be extrapolated to other types of attacks or systems with distinct neural architectures. Building ML systems that are resilient to a variety of threat models is therefore still an unresolved problem.

One future research direction can be identifying robust features for ML systems. Recent research such as [Tong et al. 2019] has shown that the collection of conserved features –

those that cannot be modified unilaterally without compromising malicious functionality – is the key to ML robustness against realizable attacks. On the other hand, the vulnerability of ML systems to adversarial examples can be directly ascribed to the presence of *non-robust features* [Ilyas et al. 2019]. Several unresolved issues can be explored, including how to automatically identify robust features and whether robust features can be constructed from non-robust features, such as via non-linear combination.

Bibliography

A. Athalye, N. Carlini, and D. Wagner. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, pp. 274–283.

B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Srndic, P. Laskov, G. Giacinto, and F. Roli. 2013. Evasion attacks against machine learning at test time. In *European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 387–402.

N. Carlini and D. A. Wagner. 2017. Towards evaluating the robustness of neural networks. *IEEE Symposium on Security and Privacy*, pp. 39–57.

J. M. Cohen, E. Rosenfeld, and J. Z. Kolter. 2019. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*.

N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma. 2004. Adversarial classification. In *SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 99–108.

J. Deng, J. Guo, N. Xue, and S. Zafeiriou. 2019. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4690–4699.

K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. 2018. Robust physical-world attacks on deep learning visual classification. In *Computer Vision and Pattern Recognition*.

P. Fogla, M. Sharif, R. Perdisci, O. Kolesnikov, and W. Lee. 2006. Polymorphic blending attacks. In *USENIX Security Symposium*.

I. Goodfellow, J. Shlens, and C. Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. 2019. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*.

M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana. 2019. Certified robustness to adversarial examples with differential privacy. In *IEEE Symposium on Security and Privacy*.

B. Li and Y. Vorobeychik. 2018. Evasion-robust classification on binary domains. *ACM Transactions on Knowledge Discovery from Data*.

W. Liu, Y. Wen, Z. Yu, and M. Yang. 2016. Large-margin softmax loss for convolutional neural networks. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 507–516.

W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. 2017. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 212–220.

D. Lowd and C. Meek. 2005. Adversarial learning. In *ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pp. 641–647.

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

S.-M. Moosavi-Dezfooli, A. Fawzi, and P. 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*, pp. 2574–2582.

N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. 2016. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 372–387. IEEE.

O. M. Parkhi, A. Vedaldi, and A. Zisserman. 2015. Deep face recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 41.1–41.12.

F. Schroff, D. Kalenichenko, and J. Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823.

M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter. 2016a. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *ACM SIGSAC Conference on Computer and Communications Security*, pp. 1528–1540. ACM.

M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter. 2016b. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, p. 1528–1540. Association for Computing Machinery, New York, NY, USA. ISBN 9781450341394.

C. Smutz and A. Stavrou. 2012. Malicious pdf detection using matadata structural features. In *Annual Computer Security Applications Conference*, pp. 239–248.

N. Šrndic and P. Laskov. 2014. Practical evasion of a learning-based classifier: A case study. In *IEEE Symposium on Security and Privacy*, pp. 197–211.

Y. Sun, Y. Chen, X. Wang, and X. Tang. 2014a. Deep learning face representation by joint identification-verification. In *Advances in Neural Information Processing Systems*, pp. 1988–1996.

Y. Sun, X. Wang, and X. Tang. 2014b. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1891–1898.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. 2014. Intriguing properties of neural networks. In *International Conference on Learning Representations*.

Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. 2014. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701–1708.

L. Tong, B. Li, C. Hajaj, C. Xiao, N. Zhang, and Y. Vorobeychik. 2019. Improving robustness of ml classifiers against realizable evasion attacks using conserved features. In *USENIX Security Symposium*.

L. Tong, Z. Chen, J. Ni, W. Cheng, D. Song, H. Chen, and Y. Vorobeychik. June 2021. Facesec: A fine-grained robustness evaluation framework for face recognition systems. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13254–13263.

Y. Wen, K. Zhang, Z. Li, and Y. Qiao. 2016. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, pp. 499–515. Springer.

E. Wong and Z. Kolter. 2018. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International conference on machine learning*, pp. 5286–5295.

E. Wong, L. Rice, and J. Z. Kolter. 2020. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*.

T. Wu, L. Tong, and Y. Vorobeychik. 2020. Defending against physically realizable attacks on image classification. In *8th International Conference on Learning Representations (ICLR)*.

W. Xu, Y. Qi, and D. Evans. 2016. Automatically evading classifiers: A case study on PDF malware classifiers. In *Network and Distributed System Security Symposium*.

F. Zhang, P. Chan, B. Biggio, D. Yeung, and F. Roli. 2015. Adversarial feature selection against evasion attacks. *IEEE Transactions on Cybernetics*.