Adversarial Recommender Systems: Attack, Defense, and Advances

Vito Walter Anelli, Yashar Deldjoo, Tommaso Di Noia, and Felice Antonio Merra

1 Introduction

Machine learning (ML) models are increasingly deployed in online systems due to their ability to generalize on new data. In recommender systems (RS), latent factor models (LFM), such as matrix factorization (MF), are among the most prominent ML techniques that have been utilized in various recommendation settings. From an algorithmic point-of-view, these models try to find latent factors of users and items whose interactions can explain an unknown preference. Nearly three decades of research on RS has resulted in various recommendation models that aim to exploit the user interaction data and side-information [39, 86] to improve recommendation performance accuracy and beyond-accuracy aspect [58]. Notwithstanding their great success, inherent to machine-learned algorithms, lies the fundamental assumption of *data stationarity*, that is, both the training data and test data are sampled from a similar (and possibly unknown) distribution [17]. In an adversarial setting, however, an intelligent and adaptive attacker may deliberately manipulate data—thereby violating the stationarity assumption—aiming to undermine RS operation and compromise the system's integrity. As RS are adopted in many life-affecting decision-making scenarios, this vulnerability raises several issues whether ML-based techniques can be safely adopted in different recommendation scenarios or if they must (and can) be redesigned to exhibit desired security behavior to be trustworthy in the face of aggressive provocation from determined attackers.

Authors have equally contributed to the chapter. They are listed in alphabetical order.

V. W. Anelli · Y. Deldjoo · T. Di Noia (✉) · F. A. Merra
Polytechnic University of Bari, Bari, Italy
e-mail: vitowalter.anelli@poliba.it; yashar.deldjoo@poliba.it; tommaso.dinoia@poliba.it; felice.merra@poliba.it

© Springer Science+Business Media, LLC, part of Springer Nature 2022

F. Ricci et al. (eds.), *Recommender Systems Handbook*,
https://doi.org/10.1007/978-1-0716-2197-4_9

Table 1 Different categories of attacks and example research in each case

Attack type	Example research
Hand-engineered shilling attacks	
• Attack by leveraging interaction data	[61, 76]
• Attack by exploiting semantic data	[4]
• Studying the impact of data characteristics on shilling attacks	[38]
• Detection and defense of shilling attack	[2, 15, 20, 21, 117, 118]
Machine-learned data poisoning optimization	
• Factorization-based models	[31, 34, 45, 56, 57, 64, 67, 92]
• Reinforcement Learning models	[23, 88, 115]
• Other recommendation models	[32, 44, 108]
• Defense	[67]
Adversarial machine-learned attacks	
• Adversarial perturbations on model parameters	[42, 55, 110]
• Adversarial perturbation on content data	[5, 7, 75, 93]
• Defense, robustification, and evaluation	[10, 55, 93]

Security of RS has been studied in two different contexts in the history of RS research in the last three decades, the one related to hand-crafted shilling attacks since 2000 [19, 38, 89], the other one to machine-learned adversarial attacks starting from 2016 [34, 55, 115]. A third and recently emerging area is also ML-based approaches for shilling attacks [44, 64], which are, to some degree, similar to adversarial attacks from the viewpoint of using ML-learned techniques to alter the recommendation performance with minimal data variations. Shilling attacks utilize the similarity patterns between users' rating profiles and *manually* injected fake rating patterns identical to those already in the system such that it can push the desired item into the recommendation list of users (push attack) or recommend non-relevant items to create a mistrust on a system (nuke attack). In contrast, attacks based on adversarial machine learning (AML) focus on learning additive perturbations, that once injected into the data they can manipulate data stationary assumption and alter the recommendation results toward an engineered and often malicious outcome [37, 41]. Table 1 summarizes attack types with some corresponding references.

Despite the similarities between ML classification and recommendation learning tasks, there are considerable differences/challenges in adversarial attacks on RS compared with ML and the degree to which the subject has been studied in the respective communities:

- *Poisoning vs. adversarial attack.* As we said before, in the beginning, the main focus of the RS research community has been on *hand-engineered* fake user profiles (a.k.a. shilling attacks) against rating-based CF [38]. Given a URM with n real users and m items, the goal of a shilling attack is to augment a fraction of malicious users $\lfloor \alpha n \rfloor$ ($\lfloor . \rfloor$ is the floor operation) to the URM ($\alpha \ll 1$) in which each malicious use profile can contain ratings to a maximum number of C items.

The ultimate goal is to harvest recommendation outcomes toward an illegitimate benefit, e.g., pushing some targeted items into the top-K list of users for market penetration. Shilling attacks against RS had established literature, and their development face two main milestones: the first one—since the early 2000s—where the literature was focused on building hand-crafted fake profiles whose rating assignment follow different strategy according to random, popular, love-hate, bandwagon attacks among others [52]; the second research direction started in 2016 when the first ML-optimized attack was proposed by Li et al., [64] on factorization-based RS. This work reviews a novel type of data poisoning attack that applies the adversarial learning paradigm for generating poisoning input data. Nonetheless, given their significant impact against modern recommendation models, the research works focusing on *machine-learned adversarial attacks* against RS have recently received considerable attention from the research community.

- *CF vs. classification models.* Attacks against classification tasks focus on enforcing the wrong prediction of individual instances in the data. In RS, however, the mainstream attacks rely on CF principles, i.e., mining similarities in opinions of like-minded users to compute recommendations. This interdependence between users and items can, on the one hand, *improve robustness* of CF, since predictions depend on a group of instances, not on an individual one and, on the other hands, may cause *cascade effects*, where attacks on a single user may impact other neighbor users [34].
- *Granularity and application type.* Adversarial examples created, e.g., for image classification tasks, are empowered based on a continuous real-valued representation of image data (i.e., pixel values), but in RS, the raw values are user/item IDs and ratings that are discrete. Perturbing these discrete entities is infeasible since it may lead to changing the input semantics, e.g., loosely speaking applying $ID + \delta$ can result in a new user ID. Therefore, existing adversarial attacks in the field of ML are not transferable to the RS problems trivially. Furthermore, in the context of Computer Vision (CV)—attacks against images—the perturbations often need to be "human-imperceptible" or "inconspicuous" (i.e., may be visible but not suspicious) [104]. How we can capture these nuances for designing attacks in RS remains an open challenge.

In the following, we present the outline of the current book chapter: in Sect. 2, we present foundation concepts to adversarial machine learning (AML) by presenting the common goal of empirical risk minimization (ERM) in supervised learning ML task, contrast it with the adversarial perceptive and present the countermeasure strategies. Afterward, we review the widely adopted attack and defense strategies built on top of the ERM problem. Then, in Sect. 3, we present state of the art in adversarial attack and defense against recommendation models. We also present machine-learned data poisoning attacks. In Sect. 4, we present evaluation protocols of adversarial attacks and defenses in recommendation tasks. Finally, in Sect. 5, we conclude the book chapter and summarize new arising challenges.

Please also note that the previous edition of this book contained a chapter named "Robust collaborative recommendation" [19] that is at some level relevant to the current book chapter since both address security issues in RS. However, in [19] the focus is on hand-crafted shilling attacks, whereas in the current book chapter, we focus primarily on modern machine-learned attacks based on the AML paradigm.

2 Foundations

Adversarial attacks were first investigated in 2013 in [90] by Szegedy et al. They discovered that, given an image, when added some carefully selected perturbations that are barely perceptible to the human eye, a well-trained deep neural network (DNN) could misclassify the adversarial image with high confidence. For instance, the attacker may perturb pixels of a pandas image not to be perceived by a human observer and obtain gibbon as the classification result with high confidence. These outcomes were strikingly stunning since it was expected that state-of-the-art DNNs generalize well on unknown data and do not alter the class of a given test image that is marginally perturbed using a cheap analytical approach. Szegedy et al. coined the term 'adversarial examples' and presented an optimization-based system using box-constrained L-BFGS to learn such perturbations. At this time, it was believed, as suggested by Szegedy et al., that *non-linearity* of neural networks is the main reason for their adversarial vulnerability.

A year later, Goodfellow et al. [50] proposed a counterintuitive hypothesis, informing *linearity* of neural networks—instead of their non-linearity—as the main reason for adversarial behavior. This claim, which is commonly known as the 'linearity hypothesis' in the literature, was supported by the fact that the design of neural networks intentionally encourages linear behavior, especially when using activation functions like Relu MaxOut. In other words, although these functions make the models theoretically non-linear, they are trained to function in the linear region of the activation function to counter phenomenons like the vanishing of gradients. To support this hypothesis, the authors demonstrated that the FGSM attack that worked based on the linearity assumption of neural networks was sufficient to fool deep neural networks, supporting their argument that neural networks behave like a linear classifier.

As attack strategies introduced in this chapter work primarily on classification tasks, we discuss the foundation concepts for a classification problem to keep this chapter self-contained. In a classical *supervised learning* setting, we are given a paired training dataset $\mathscr{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ where $\mathbf{x}_i \in \mathscr{X} \subseteq R^d$ is a feature vector in the *input space* $\mathscr{X}$ and $y_i \in \mathscr{Y}$ is the corresponding label in some *output space* $\mathscr{Y}$. For instance, in binary classification $\mathscr{Y}$ is binary and used as $\mathscr{Y} = \{-1, +1\}$. Each pair in $\mathscr{D}$ is assumed to be generated i.i.d.[1] from an unknown distribution P,

[1] Independent and identically distributed (i.i.d).

i.e., $(\mathbf{x},\ y) \sim P$. We also assume that we are given a suitable loss function $\mathscr{L}(.,.)$, for instance the cross-entropy loss for a neural network. The goal is to find a good candidate function $f(\mathbf{x};\theta)$ that minimizes the following empirical risk

$$\min_{\theta}\ \mathbb{E}_{(\mathbf{x},y)\sim P}\ \mathscr{L}(f(\mathbf{x};\theta),\ y) \tag{1}$$

where $\mathbb{E}_{(\mathbf{x},y)\sim P}$ is commonly termed *expected risk* of the classifier, θ is the model parameter and y is the class label for the input sample $\mathbf{x}$. As P it is often unknown, we use $\mathscr{D}$ in order to learn the suitable candidate function $f(\mathbf{x},\theta)$. The training objective function can be formulated as the following optimization problem,

$$\min_{\theta} \sum_{(\mathbf{x}_i,y_i)\in\mathscr{D}} \mathscr{L}(f(\mathbf{x}_i;\theta),\ y_i) \tag{2}$$

where $f(\mathbf{x}_i;\theta)$ and y_i are the predicted and class label for the sample i.

2.1 Adversarial Perspective

Empirical risk minimization (ERM) has been found a powerful means to tune classifiers with small population risk. Regretfully, ERM is not capable of producing models that are robust against adversarially crafted samples. In an adversarial setting, we can define a general form of the objective for adversarial attacks based on Eq. 2, which aims to force a trained model to make a wrong prediction under a minimal perturbation budget. The problem of adversarial attacks can be formally stated as follows.

Definition 1 Given a learned classifier $f(\mathbf{x};\theta)$ and an instance from the dataset $(\mathbf{x},\ y)\ \in\ \mathscr{D}$, the attacker takes the sample $\mathbf{x}$ and produces an adversarial version $\mathbf{x}_{adv} = \mathbf{x} + \delta$ such that $f(\mathbf{x}_{adv};\theta) \neq f(\mathbf{x};\theta)$. The attacker aims to do this within a minimal perturbation budget, i.e., $\|\delta\|_p \leq \epsilon$ where $\|.\|_p$ is the p-norm. The attacker's objective can then be formally stated as follows,

$$\arg\max_{\delta}\ \mathscr{L}(f(\mathbf{x}+\delta;\theta),\ y),\quad s.t.,\ \|\delta\|_p \leq \epsilon, \tag{3}$$

where ϵ is the perturbation budget, typically chosen as a small value.

■

2.2 Taxonomy of Adversarial Attacks

There have been several proposals to categorize attacks against ML algorithms. The most prominent categorization is based on the following dimensions: *timing*, *knowledge*, and *goal*. We use these dimensions as the basis to distill the main characteristics of adversarial attacks and introduce a taxonomy of attack systems that highlights the various aspects of this area.

2.2.1 Attack's Timing

The first crucial aspect in modeling attacks is *when* they occur in the learning pipeline of the ML system. This consideration gives rise to a dichotomy, which is central to attacks on ML models: *attacks on models*—or, more precisely, decisions made by the learned model—and attacks on *algorithms*, occurring before model training by modifying part of the training data used for training [98]. These two categories are respectively known as *evasive* and *data poisoning* attacks:

- **Evasive attack:** The attacks aim to avoid detection—or evade the decisions made by the learned model, thus the name evasive—by directly manipulating malicious test samples. Therefore, these attacks occur after the ML model is trained or in the inference (test) phase. The model is fixed, and the attacker cannot alter the model parameter or structure.
- **Poisoning attack:** These attacks happen before the ML model is trained. The attacker can add false data points (or *poisons*) into the model training data, causing the trained model to produce an erroneous prediction. Poisoning attacks have been explored in the literature for a variety of tasks [98], such as (1) attacks on binary classification for tasks such as label flipping or against kernelized SVM, (2) attacks on unsupervised learning such as clustering and anomaly detection and, (3) attacks on matrix completion task in RS.

2.2.2 Attacker's Knowledge

When modeling attacks, the second important consideration is what–or how much–information the adversary has about the learning model, the algorithm, or the training data they aim to attack. This distinction leads to the following classification: *white-box*, *black-box*, and *gray-box* attacks.

1. **Perfect-knowledge (PK) adversary**: A perfect-knowledge (or *white-box*) adversary assumes that the attacker has precise information about the learned model (the actual classifier), including, e.g., the features, the learning algorithm, hyper-parameters, among others. The adversary can subsequently adjust the attack strategy to account for the defense. In the field of cybersecurity, it has been shown that assuming attacker having no knowledge—or security by obscurity—

is ineffective [35]; on the opposite hand, if a defender can be robust to PK attacks, it will surely be robust to more knowledge-limited attacks; thus these threat models offer natural reasons for consideration. Therefore, a PK attack is the most potent possible threat model.

2. **Limited-knowledge (LK) adversary**: A limited-knowledge (or *gray-box*) adversary has some, albeit incomplete, level of knowledge about the system under attack. For instance, the attacker may know the classifier (or its type) or the training data used, but not simultaneously. However, it is assumed that the adversary can collect and build a surrogate dataset $D' = \{\mathbf{x}'_i, y'_i\}_{i=1}^{n}$ from the same distribution p from which $\mathscr{D}$ was drawn. This dataset can be used to train a classifier that should be similar to the actual defender in place [46].
3. **Zero-knowledge (ZK) adversary**: A zero-knowledge (or *black-box*) adversary has no information about the learned model or the algorithm used by the learner before developing the attack.

For clarification, we present a simplifying description of the above threat models for adversarial attacks, given by Biggio et al. in [16]. Let f be a learned unsecured model and d a detector component, for instance a classifier for anomaly detection, implemented to secure f. Possible scenarios for the previous classification are depicted below:

1. A PK attacker gets full knowledge of the victim model f and its security granted by the detector d, knows the model parameters of d, and uses this information to craft adversarial samples to evade d and corrupt f.
2. A LK attacker is aware f is being secured with a detection component d, knows the training scheme of both models, but his knowledge is *limited* by a denied access to the detector and the victim model (or the exact training data). In this scenario, the adversary may not be able to craft effective samples that thoroughly address his malicious goal.
3. A ZK adversary is assumed to generate adversarial examples to attack the unsecured model f being not aware that a detector d is in place. d will successfully protect f if it can detect and reject all the adversarial samples.

2.2.3 Attacker's Goals

While the attacker's objective to execute an attack on ML systems may encompass a broad spectrum of possible goals, we can distill attacks into two major classes in terms of the attacker's goals: **targeted attacks** and **untargeted (reliability) attacks**. In both attack types, the attacker's main attempt is to maximize mistakes in the learning algorithm's decisions with respect to the ground truth.

Definition 2 (Targeted Adversarial Attack) Given a trained classifier $f(\mathbf{x}; \theta)$ and a test instance from the dataset $\mathbf{x}_0 \in \mathscr{D}$ where $f(\mathbf{x}_0; \theta) = y_0$, the goal of the attacker is to apply a change in the label for $\mathbf{x}_0$ to a specific target label $y_T \neq y_0$, known as *misclassification label*. We can formulate the problem as

$$\begin{aligned} \min_{\delta} \quad & \|\delta\| \\ \text{s.t.:} \quad & f(\mathbf{x}_0 + \delta; \theta) = y_T \end{aligned} \tag{4}$$

Note that for images, a second constraint such as $\mathbf{x}_0 + \delta \in [0, 1]^n$ is used, to impose a value-clipping constraint, where its goal is to bound the adversarial samples into a predefined range so that the perturbation remains human-imperceptible. Alternatively, the above problem can be expressed in an unconstrained optimization problem formulation

$$\min_{\delta:\|\delta\| \leq \epsilon} \quad \mathscr{L}(f(\mathbf{x}_0 + \delta; \theta),\ y_T) \tag{5}$$

One can note that in this case, the attacker aims to *minimize* the loss between adversarial prediction and the misclassification label y_T.

■

Definition 3 (Untargeted Attack) The goal of the attacker in untargeted attack is to induce any misclassfication, such that

$$\begin{aligned} \min_{\delta} \quad & \|\delta\| \\ \text{s.t.:} \quad & f(\mathbf{x}_0 + \delta; \theta) \neq y_0 \end{aligned} \tag{6}$$

where $\mathbf{x}_0 \in \mathscr{D}$ is a test instance and y_0 is the true class label, such that $f(\mathbf{x}_0; \theta) = y_0$. Similarly, here also we can formulate the untargeted adversarial attack as an unconstrained optimization problem where the goal of the attacker is to *maximize* the loss between the adversarial term and y_0

$$\max_{\delta:\|\delta\| \leq \epsilon} \quad \mathscr{L}(f(\mathbf{x}_0 + \delta; \theta),\ y_0) \tag{7}$$

■

2.3 Adversarial Robustness: A Unified View of Adversarial Attacks and Defenses

In this section, we aim to study the adversarial defenses of ML systems. We start by presenting an optimization view of *adversarial defense*—or more precisely *adversarial robustness*—through the lens of robust optimization and use a min-max problem formulation to capture the notion of security against adversarial attacks in a principled manner. This formulation allows us to be precise about the type of security guarantee we would like to achieve, i.e., the broad class of attacks we want to resist. The empirical risk minimization (ERM) formulation introduced in

Eq. 1 does not yield models that are robust against adversarial examples. To reliably train models that are robust to adversarial attacks, it is necessary to redefine the ERM paradigm appropriately. In adversarial supervised learning, we assume that the adversary can modify training data distribution in a particular manner. For example, the adversary can adjust the input feature vectors to cause prediction errors.

Definition 4 (Adversarial Empirical Risk Minimization (AERM)) Given a trained classifier $f(\mathbf{x}; \theta)$ and a test instance sampled from the distribution $(\mathbf{x}, y) \sim P$ that the classifier has been trained on, the attack model $x_{adv} = A_f(\mathbf{x}; \theta)$ takes $\mathbf{x} \in \mathbb{R}^d$ and projects it into data in the adversarial target set $Z \subset \mathscr{X} \times \mathscr{Y}$ aimed at increasing the prediction loss. The resulting AERM problem can be formulated as

$$\min_{\theta} \; p(\theta), \text{ where } \; p(\theta) = \mathbb{E}_{(\mathbf{x},y)\sim P} \left[\max_{\mathbf{x}_{adv}} \mathscr{L}(A_f(\mathbf{x}; \theta), y) \,|(\mathbf{x}, y) \in Z \right] \tag{8}$$

where $\theta \in \mathbb{R}^d$ is the model parameter associated with f. The most commonly studied attack model so far is the *additive* adversarial perturbation in the form $\mathbf{x}_{adv} = \mathbf{x} + \delta$, where $\delta \in S \subseteq \mathbb{R}^d$ is the adversarial perturbation taken from the allowed adversarial set S. In addition, since P is unknown, therefore we use the training dataset $\mathscr{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ as a proxy, thus

$$\min_{\theta} \; p(\theta), \text{ where } \underbrace{p(\theta)}_{\text{adversarial loss}} = \sum_{(\mathbf{x}_i, y_i)\in\mathscr{D}} \underbrace{\left[\max_{\delta \in S} \mathscr{L}(f(\mathbf{x}_i + \delta; \theta), \; y_i) \right]}_{\text{worst-case prediction loss } \mathscr{L}} \tag{9}$$

commonly represented in the following **min-max formulation** to capture the notion of security against adversarial attacks in a principled approach

$$\underbrace{\min_{\theta} \sum_{(\mathbf{x}_i, y_i)\in\mathscr{D}} \underbrace{\left[\max_{\delta : \|\delta\| \leq \epsilon} \mathscr{L}(f(\mathbf{x}_i + \delta; \theta), \; y_i) \right]}_{\text{worst-case loss } \mathscr{L} = \textbf{optimal attack}}}_{\textbf{robust classification} \text{ against adversarial attack}} \tag{10}$$

where $\delta : \|\delta\| \leq \epsilon$ represents the adversarial perturbation bounded by ϵ, or the perturbation budget.

■

The AERM problem defined above can be viewed as a composition of an **inner maximization attack** and **outer minimization defensive** problem. The inner maximization is the definition of adversarial attack presented in Sect. 2.1, which aims to find perturbation δ that maximizes the prediction loss of the model f. The outer minimization aims to find model parameters that minimize the adversarial loss produced by the inner attack problem or, more precisely, robust optimization using adversarial training.

2.4 Definition of Adversarial Attack and Countermeasure Strategies

In this section, we aim to formally introduce and define the most prominent attack and defense strategies used in ML systems commonly used in image classification task. The subsequent sections will discuss the same type of attack and defense strategies used against RS methods, with adaptation techniques to make them usable for recommendation tasks.

2.4.1 Attack Models

Various adversarial attack methods that aim to find a non-random perturbation δ to produce an adversarial example $x_{adv} = x + \delta$ that can cause an erroneous prediction (e.g., misclassification) are formally presented in this section. These attack methods all aim to solve the inner-maximization problem in Eq. 10 or more precisely the targeted and untargeted attack problems given by Eqs. 5 and 7.

Definition 5 (Limited-Memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS)) The L-BFGS attack is the crucial work studied by Szegedy et al. [90], the one that captured researchers' attention the first time on the vulnerability of DNNs for image recognition tasks. Its main goal is to find a minimally distorted adversarial example based on a l_2 distance using an intuitive box-constrained optimization problem.

$$\begin{aligned} &\min_{\delta} \ \|\delta\|_2^2 \\ &\text{s.t.:} \ \ f(\mathbf{x}+\delta) = y_T \ \ \text{and} \ \ \mathbf{x}+\delta \in [0, M]^n \end{aligned} \tag{11}$$

where n is the number of features, M is the maximum value among the pixels in the image, and y_T is the misclassification label. From an implementation point of view, the above problem has two constraints and is hard to solve. The following equivalent with one constraint is solved instead.

$$\begin{aligned} &\min_{\delta} c \cdot \|\delta\|_2^2 + \mathscr{L}(f(\mathbf{x};\theta), y_T) \\ &\text{subject to} \ \ \mathbf{x}+\delta \in [0, M]^n \end{aligned} \tag{12}$$

the constraint $\mathbf{x}+\delta \in [0, M]^n$ is addressed by utilizing a box-constrained optimizer and a line-search that finds the best parameter c.

■

Definition 6 (Fast Gradient Sign Method (FGSM)) The FGSM attack model [50] was originally designed to exploit the 'linearity' of DNNs in the higher dimensional space. The goal of Goodfellow et al. [50] was to solve Eq. 6

(untargeted attack) by adding arbitrary perturbation to the original clean input with the ℓ_∞-bound constraint (i.e., $||\delta||_\infty \leq \epsilon$) such that the training loss of the target model increases, thus reducing classification confidence and improving the likelihood of inter-class confusion. While there is no guarantee that increasing the training loss by a certain amount will yield misclassification, this is nevertheless a sensible direction to exercise since the prediction error of a wrongly classified sample is, by definition, larger than the correctly classified one. The key idea in *untargeted FGSM* is to use a first-order approximation of the loss function and utilize the sign of the gradient function to construct adversarial samples for the adversary's target classifier f, obtaining.

$$\mathbf{x}_{adv} = \mathbf{x} + \epsilon \cdot \text{sign}(\nabla_x \mathscr{L}(f(\mathbf{x}; \theta),\ y)) \tag{13}$$

where ϵ (perturbation level) represents the attack strength and ∇_x is the gradient of the loss function w.r.t. input sample $\mathbf{x}$, y is the correct label and sign(.) is the sign operator. The corresponding approach for *targeted FGSM* [60] is

$$\mathbf{x}_{adv} = \mathbf{x} - \epsilon \cdot \text{sign}(\nabla_x \mathscr{L}(f(\mathbf{x}; \theta),\ y_T)) \tag{14}$$

where y_T is the target misclassification class label for sample $\mathbf{x}$.

■

Several variants of the FGSM has been proposed in the literature [28, 104]. For instance, the fast gradient value (FGV) method [82], which instead of using the sign of the gradient vector in FGSM, uses the actual value of the gradient vector to modify the adversarial change, or basic iterative method (BIM) [60] (a.k.a. iterative FGSM) that applies FGSM attack multiple times *iteratively* using a small step size and within a total acceptable input perturbation level, according to

$$\mathbf{x}_{adv}^{t+1} = FGSM(\mathbf{x}_{adv}^{t}) \tag{15}$$

Projected gradient descent (PGD)[68] is similar to BIM attack. It starts from a random position in the clean image neighborhood.

Definition 7 (Carlini and Wagner (CW-ℓ_0, CW-ℓ_2, CW-ℓ_∞)) Carlini and Wagner is a powerful attack model for finding adversarial perturbation under three various distance metrics (ℓ_0, ℓ_2, ℓ_∞). Its key insight is similar to L-BFGS as it transforms the constrained optimization problem into an empirically chosen loss function to form an unconstrained optimization problem as

$$\min_{\delta} \left(\|\delta\|_p^p + c \cdot h(\mathbf{x} + \delta, y_T) \right) \tag{16}$$

where $h(\cdot)$ is the candidate loss function. ■

The C&W attack has been used with several norm-type constraints on perturbation l_0, l_2, l_∞ among which the l_2-bound constraint has been reported to be most effective [25, 26, 29]. The CW-ℓ_2 problem formulation for a targeted attack aiming is given by

$$\begin{aligned} &\min_{\delta} \left(\|\mathbf{x}_{adv} - \mathbf{x}\|_2^2 - c \cdot h(\mathbf{x}_{adv}, y_T) \right) \\ &h(\mathbf{x}_{adv}) = \max \left(\max_{i \neq t} Z\{\mathbf{x}_{adv_i}\} - Z\{\mathbf{x}_{adv_t}\}, -K \right) \\ &\mathbf{x}_{adv} = tanh(arctanh(\mathbf{x}) + \delta) + 1)) \end{aligned} \tag{17}$$

where Z(x) denotes the logit corresponding to i-th class. By increasing the classification confidence K, the adversarial sample will be misclassified with higher confidence.

2.4.2 Adversarial Countermeasure Strategies

From a broad perspective, the defensive mechanism against adversarial attacks can be classified into one of the following approaches:

- **Increasing robustness of learning model:** These methods aim to formulate models that can correctly classify both adversarial and clean samples. At their heart, many of these methods attempt to create models less sensitive to irrelevant data variations, e.g., by regularizing models to mitigate the attack surface and bound responsiveness to samples that lie off the data manifold.
- **Detection:** While a robust classifier correctly labels adversarial perturbed samples, robustness may be alternatively achieved by *detection of adversarial examples*. A sample that is detected as an adversarial example is rejected.

Note that a recurring hypothesis about the cause for adversarial samples is that these examples lie off the data manifold and are sampled from a different distribution [84]. The learning model has no exposure to such off-manifold regions during training time, and hence its behavior can be controlled arbitrarily. While robust classification aims to map the data back to on-manifold data (e.g., natural image manifold) and recover its actual label, the detection methods treat the problem as an *anomaly detection problem*, only requiring to determine whether the input is an on-manifold data or reject it otherwise.

Increasing Robustness of Learning Model

For what concerns the robustness of the learning model, we discuss two categories of algorithms for classification task:

1. **Robust optimization:** This is a theoretically grounded framework, as it aims to integrate robustness into learning. As such, robust optimization provides the means to guarantee or certify robustness in a principled manner. We discuss

adversarial training as one of the most common approaches for robust optimization against adversarial attacks here.
2. **Distillation** this is a heuristic approach for making gradient-based attacks more difficult to execute by effectively rescaling the output function to ensure that gradients become unstable [42].

Robust Optimization and Adversarial Training
In Sect. 2.3, we presented a unified framework for adversarial attack and defense by showing a zero-sum game between the learning model, in which we can formulate adversarial robustness as a *robust optimization* problem that seeks to find a solution for the worst-case input perturbation with respect to the set of allowed perturbation (i.e., $\delta \in S$)

$$\min_{\theta} \sum_{(\mathbf{x}_i, y_i) \in \mathscr{D}} \max_{\delta \in S} \mathscr{L}(f(\mathbf{x}_i + \delta; \theta),\ y_i) \tag{18}$$

where $\mathscr{L}$is the loss function, f is the learning model characterized by the parameter θ, and $(\mathbf{x}_i, y_i) \in \mathscr{D}$ is a training sample. Typically $\|\delta\|_p \leq \epsilon$ known as the adversarial budget is used to represent the set S where $\|\cdot\|_p$ is the l_p norm.

Based on the above intuition, adversarial regularization, also known as *adversarial training* (AT) was proposed by Goodfellow et al. [50] for formulating a robust classifier. AT is a data augmentation training process, where one tries at each update to approximately solve the inner attack problem in Eq. 18, generating adversarial examples and injecting them into training data. Assuming worst-case loss function to be

$$\mathscr{L}_{wc}(f(\mathbf{x} + \delta; \theta), y) = \max_{\delta : \|\delta\| \leq \epsilon} \mathscr{L}(f(\mathbf{x} + \delta; \theta), y)$$

AT utilizes $\mathscr{L}_{wc}$ as a regularization component to explicitly trade-off between robustness (on adversarial samples) and accuracy (on non-adversarial data).

$$\min_{\theta} [\mathscr{L}(f(\mathbf{x}; \theta), y) + \lambda \mathscr{L}_{wc}(f(\mathbf{x}; \theta), y)] \tag{19}$$

■

where $0 < \lambda < 1$ is the regularization coefficient, controlling the amount of trade-off. The robustness achieved by AT strongly depends on the strength of the adversarial examples used. For instance, training on fast non-iterative attacks such as FGSM may yield robustness against non-iterative attacks, and not against PGD attacks [60, 85]. Madry et al. in [68] showed that training on multi-step PGD adversaries achieves state-of-the-art robustness levels against l_∞ attacks.

Adversarial Training of BPR-MF

BPR is the state-of-the-art method for personalized ranking implicit feedbacks. The main idea behind BPR is to maximize the distance between positively and negatively rated items. Given the training dataset D composed of positive and negative items for each user, and the triple (u, i, j) (user u, a positive item i and negative item j), the BPR objective function is defined as

$$\mathscr{L}_{BPR}(\mathscr{D}|\Theta) = \arg\max_{\Theta} \sum_{(u,i,j)\in\mathscr{D}} ln\, \sigma(\hat{x}_{ui}(\Theta) - \hat{x}_{uj}(\Theta)) - \lambda \|\Theta\|^2 \tag{20}$$

where σ is the logistic function, and $\hat{x}_{ui}$ is the predicted score for user u on item i and $\hat{x}_{uj}$ is the predicted score for user u on item j; $\lambda \|\Theta\|^2$ is a regularization method to prevent over-fitting.[2] Adversarial training of BPR-MF similar to Eq. 19 can be formulated as

$$\mathscr{L}_{APR} = \underbrace{\min_{\theta} \sum_{(u,i,j)\in D} [\mathscr{L}_{BPR}(\mathscr{D}|\Theta) + \lambda \underbrace{\max_{\delta:\|\delta\|\leq\epsilon} \mathscr{L}_{BPR}(\mathscr{D}|\Theta+\delta)}_{\text{optimal attack model against BPR}}]}_{\text{optimal robustness preserving defensive}} \tag{21}$$

■

3 A Classification of Adversarial Attacks on Defenses of RS

The majority of the attack strategies that would be introduced in this chapter were initially conceived in the computer vision (CV) domain for the image classification task. Adversarial examples created for images, however, are empowered given the continuous real-valued nature of image data, i.e., pixel values, but the input data of recommender models are mostly discrete features (i.e., user ID, item ID, and other categorical variables) [55]. Perturbing these discrete features by applying noise is meaningless since it may change their semantics. Therefore, existing adversarial attacks in the field of CV do not apply to the RS problems trivially. Instead, they are applied at a deeper level—e.g., at the level of their intrinsic model parameters rather than the extrinsic inputs [55].

In the current section, we adopt a pragmatic approach to classify the research articles conducting adversarial attacks against RS and discuss each category's attack

[2] As it can be noted, BPR can be viewed as a classifier on the triple (u, i, j), where the goal of the learner is to classify the difference $\hat{x}_{ui} - \hat{x}_{uj}$ as correct label +1 for a positive triple sample and 0 for a negative instance.

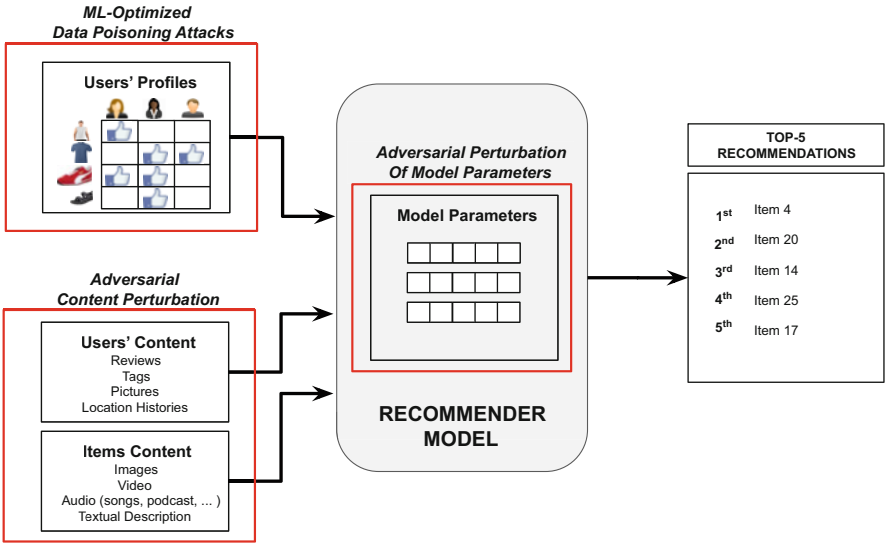

Fig. 1 A notional view of the possible injection adversarial perturbations on (**a**) users profiles, (**b**) their content data, and (**c**) the learned model parameters. Depending on where adversarial perturbations have been applied, different adversarial attack types, as mentioned in the upper part of Table 2 have been proposed in the literature

and defensive strategies. Our classification entails categorization of attacks based on their *level of granularity* according to:

- Adversarial perturbation of model parameters;
- Adversarial perturbation of content;
- ML-optimized data poisoning attacks.

Figure 1 shows a schematic representation of the three possible points of adversarial perturbations. While the perturbation of model parameters belongs to evasion attacks (decision-time attacks), the URM's perturbation and the content data are considered poisoning attacks (data poisoning attacks) since they are added before the model's training. Moreover, to provide an overview, Table 2 introduces the adversarial attacks, which have been used over the last few years in RS research. It highlights the reviewed research articles according to the following dimensions:

- **Approach.** This column lists the authors' approach name to the proposed adversarial attack/defensive mechanism and provides the reference to the main paper.
- **Attack Model.** This column represents the main *attack* methods applied on various recommendation models. They include *FGSM* and *CW* (cf. Sect. 2.4.1), and generative attacks based on generative adversarial networks (GANs).
- **RS Model.** Given the machine-learned (optimization)-based approach for most of the considered papers, this column represents the core recommendation

Table 2 Classification of approaches that use adversarial learning for attacking and defending RS models or ML-optimized poisoning attack

Category	Approach	Attack model	RS model	Defense
AML Model parameter perturbation	APR [55]	FGSM	LFM	Adv.Train.
	FGACAE [110]	FGSM	AE	Adv.Train.
	ACAE [109]	FGSM	AE	Adv.Train.
	FNCF [42]	FGSM	NN	Distillation
	AdvIR [78]	FGSM	LFM	Adv.Train.
	AMASR [95]	FGSM	NN	Adv.Train.
	ATMBPR [99]	FGSM	LFM	Adv.Train.
	SACRA [65]	FGSM	LFM	Adv.Train.
	RAP [13]	Generative	LFM	Adv.Train.
AML Content feature perturbation	AMR [93]	FGSM	LFM	Adv.Train
	TAaMR [75]	FGSM, BIM, PGD	LFM	×
	VAR [5]	FGSM, PGD, C&W	LFM	Adv.Train.
ML-optimized Poisoning Attack	S-attack [45]	Hit Ratio maximization	LFM	SVM Class.
	LOKI [115]	Reinforcement Learning	Black-box	×
	PoisonRec [88]	Reinforcement Learning	Black-box	×

prediction function according to LFM and non-linear models such as the ones based on *auto-encoder* (AE) and *neural network* (NN).

- **Defense Approach.** This column characterizes the countermeasure defensive approach against adversarial attacks. We have found that *adversarial training (a.k.a. adversarial regularization)* as the most adopted defensive approach irrespective of the attack model, while *distillation* being adopted only by a single paper [42].

3.1 Adversarial Perturbation of Model Embedding Parameters

While the attacks presented in Sect. 2.4.1 commonly evaluate the effectiveness of perturbation generated on loss function used for image classification task such as the cross-entropy of a neural network, in the case of RS, these strategies have been re-adapted in order to learn perturbation with respect to the loss function specific to RS tasks. For instance, using the matrix factorization (MF) model trained with BPR (known as MF-BPR)—the state-of-the-art ranking-based criterion for item recommendation—several works have investigated the robustness of embedding parameters added to user embeddings $p_u = p_u + \delta$, item embeddings $q_i = q_i + \delta$ or both in which $p_u, q_i \in \mathbb{R}^F$ are F-dimensional embedding factors.

Attacks

One of the first adversarial attack strategies against the recommender system's parameters was described by He et al. [55]. The authors studied the robustness of

BPR-MF [80] to adversarial perturbation on the users and items representation in the latent space. Their attack, built on the **FGSM** by Goodfellow et al. [50], was the first proposal to address adversarial perturbation in recommender systems formally. The proposed FGSM-based strategy approximates $\mathscr{L}$ by linearizing it around an initial zero-matrix perturbation Δ_0 and applying the max-norm constraint. The adversarial perturbation Δ^{adv} is defined as:

$$\Delta^{adv} = \epsilon \frac{\Pi}{\|\Pi\|} \quad \text{where} \quad \Pi = \frac{\partial \mathscr{L}(\Theta + \Delta_0)}{\partial \Delta_0} \tag{22}$$

where $||\cdot||$ is the l_2-norm. After the calculation of Δ^{adv}, He et al. added this perturbation to the current model parameters $\Theta^{adv} = \Theta + \Delta^{adv}$ and generated the recommendation lists with this perturbed model parameter. He et al. in [55] demonstrated that perturbation obtained from the FGSM with $\epsilon = 0.5$ can impair the accuracy of item recommendation by an amount equal to -26.3%. Inspired by the effectiveness of this attack, several works have performed a similar perturbation against different recommender approaches such as visual-based recommender [91], tensor-factorization machines [30], deep-learning models [109, 110].

Similar to the advances in CV, the single-step FGSM attack has been extended with an iterative strategy in the recommender domain. The authors of [6] proposed an FGSM-based *iterative* strategy to create more effective ϵ-clipped perturbations. The initial model parameters are defined as

$$\Theta_0^{adv} = \Theta + \Delta_0 \tag{23}$$

Starting from this initial state of model parameters, the authors introduce an element-wise clipping function $Clip_{\Theta,\epsilon}$ to limit the perturbation of each original embedding value inside the $[-\epsilon, +\epsilon]$ interval, a step size α which is the maximum perturbation budget of each iteration. The first iteration ($k = 1$) is then defined by:

$$\Theta_1^{adv} = Clip_{\Theta,\epsilon}\left\{\boldsymbol{\Theta}_0^{adv} + \alpha \frac{\Pi}{\|\Pi\|}\right\} \text{ where } \Pi = \frac{\partial \mathscr{L}(\Theta + \Delta_0)}{\partial \Delta_0} \tag{24}$$

and they generalized the k-th iteration of the K-iterations multi-step attack as:

$$\Theta_k^{adv} = Clip_{\Theta,\epsilon}\left\{\boldsymbol{\Theta}_{k-1}^{adv} + \alpha \frac{\Pi}{\|\Pi\|}\right\} \text{ where } \Pi = \frac{\partial \mathscr{L}(\Theta + \Delta_{k-1}^{adv})}{\partial \Delta_{k-1}^{adv}} \tag{25}$$

where $k \in [1, 2, \ldots, K]$, Δ_k^{adv} is the adversarial perturbation at the k-th iteration, and Θ_k^{adv} is the sum of the original model parameters Θ with the perturbation at the k-th iteration. Inspired by the advances of AML in iterative attacks, they considered

two different versions of multi-step optimized adversarial perturbation: the Basic Iterative Method (BIM) [59] and the Projected Gradient Descent (PGD) [68] approaches. The authors have demonstrated the iterative adversarial strategies to be considerably more effective than the single-step FGSM method such that a state-of-the-art model-based recommender model (i.e., BPR-MF) can be impaired/weakened so much so that their performance become worse than a random recommender (a more than 90% of degradation of accuracy measures). It is possible to verify the drastic performance reduction generated by these iterative strategies against others state-of-the-art model-based recommenders similarly to the application of the single-step strategy in several models [30, 91, 109, 110].

The C&W attack has also been adapted to recommender systems by Du et al. [42], showing how it may contaminate the model performance in the testing phase. Similar to the previous strategies, the authors slightly changed the C&W approach to adapt to a recommender model (i.e., NCF in their experiments [54]). The C&W optimization problem is formulated as:

$$\begin{aligned} \min_{\Delta^{adv}} \quad & ||\Delta^{adv}|| \\ \text{s. t.} \quad & f(\Theta + \Delta^{adv}) > 0.5 \end{aligned}$$

where $f(\cdot)$ is the prediction function to mark an item relevant to a user. The authors demonstrated that the attacks got a Success Rate close to 100% in inverting the predicted importance ($f(\cdot)$) of each user-item pairs.

It is clear from the above-summarized research that model-based recommenders are highly vulnerable to limited adversarial perturbations applied to model parameters. An attacker may access the model and completely misuse a recommender's utility by slightly perturbing their latent factors. Furthermore, while these settings may be complicated to be present in a real scenario, previous attacks have also demonstrated another important aspect of model-based recommenders: the instability of the training. In fact, authors [55] have claimed that the weakness of these perturbations needs particular study and attention by researchers and practitioners. The loss of a considerable part of accuracy within such small perturbations might be generated in a real scenario with few real (benevolent) users that, with their actions, are causing a model update that will get a great negative change in performance.

Defenses

The two defense mechanisms previously described have been proposed as a defensive solution also in the recommendation settings. Respectively, the Adversarial Personalized Ranking (APR) by He et al. [55], is the first method that modified the classical loss function of a recommender model (i.e., BPR-MF) by integrating the adversarial training procedure, and the stage-wise hints training by Du et al. [42], which has been inspired by the defensive distillation of deep neural networks.

To protect against FGSM attack, He et al. [55] proposed an adversarial training strategy to make the model robust to parameters perturbations. Similar to the adver-

sarial training procedure proposed in the computer vision field [50], the authors proposed an adversarial training procedure by adding an adversarial regularization term.

The application of adversarial training has been demonstrated to make the model more robust to FGSM attacks. For instance, He et al. [55] demonstrated a reduction of attack effectiveness on BPR-MF by more than 90%. This training procedure has been applied to several recommender models, e.g., BPR-MF (AMR) [55], VBPR (AMR [91]), CDAE (FG-ACAE [109, 110]), PITF [79] (ATF [30]), etc. Furthermore, the generalization ability of the adversarially trained model that gets it less influenced by parameter perturbations may also improve accuracy performance (e.g., nDCG and HR). The adversarial training procedure by He et al. [55] has been further modified to solve different issues. Tran et al. [95] proposed a *flexible* ϵ-bounded perturbations by multiplying ϵ with the standard deviation of the perturbed parameter (Θ). Xu et al. [107] designed a *directional adversarial training* procedure to perturb the parameters towards k-nearest neighbors in the embedding space such that the directions of perturbations are bounded inside a small collaborative-aware region.

The adversarial training procedure has been applied effectively in defending the recommender systems considering only FGSM perturbations on model parameters. However, the *iterative adversarial perturbations* proposed by Anelli et al. [6] have demonstrated that the adversarial training procedure proposed by He et al. [55] fails with iterative perturbations with the same ϵ-bounded budget perturbations (e.g., accuracy performance degraded by more than 50% when the model is adversarially trained).

Another defense strategy proposed by Du et al. [42] is a form of **defensive distillation** to make a deep recommender model (i.e., NCF) more robust to C&W attacks. The stage-wise hints training procedure transfers (distills) the knowledge learned from a teacher model into a student (architectural smaller) model. The hints, or modules, are the set of parameters transferred from the teacher to the student model in the stage-wise procedure. For instance, the items and users' latent vectors can be treated as two different modules. Furthermore, the authors have integrated the student model with a *noisy layer* for increasing the robustness of parameters against the perturbations. The defense procedure has been demonstrated to reduce the success rate of the C&W attacks compared to the baseline version of the recommender.

It is important to mention that several defense strategies, as well as hundreds of adversarial attack strategies, have been designed and implemented in different domains (e.g., computer vision, speech recognition, and test processing) [104], and only a few of them have been already adapted and evaluated in recommendation tasks.

3.2 Adversarial Perturbations on Content Data

The second category of adversarial attacks against recommender systems is associated with the perturbation of content data related to users and items. Content-based and Hybrid recommender systems are the two categories of recommender that are influenced by the quality of content data. For instance, images are gaining significant success in the class of fashion, and food recommendation since the high-level features extracted from convolutional neural networks have been demonstrated to influence users' preferences. For instance, McAuley et al. have proposed different visual-based recommender models [18, 53, 70] to demonstrate how much visual features have a significant impact on users' behavior and are effective in improving the recommendation performance in cold settings. Furthermore, deep features extracted from music/video audio signal [63, 96], text features extracted from tweets [112], reviews [113], or news articles [22] demonstrate an increasing dependence of recommender models from the quality, and 'benevolence' of users and items (side) data. An immediate question would be how these models can be affected when the input data is adversarially perturbed? Are the security measures proposed to protect the feature extractor (e.g., AE, CNN, and RNN) capable of making the recommender robust to such attacks? The answer is open for further investigations.

Attacks

In this section, we will present an example of adversarial perturbation of product images used as the input of a state-of-the-art visual-based recommender system, named Visual Bayesian Personalized Ranking (VBPR) [53], which is one of the main recommenders that has been tested under adversarial settings. Tang et al. [91] have demonstrated that adding a human-imperceptible untargeted adversarial perturbation on a single product image it would likely be ranked much lower than before. Di Noia et al. [75] have investigated the application of targeted adversarial attacks on product images (i.e., FGSM and PGD) such that the CNN classifier would have classified the images of a low-recommended category of products with the category of more popular products. The authors demonstrated that an imperceptible alteration of the picture would increase more than three times the probability of a product being recommended. Liu et al. [66] proposed recommendation-aware attack strategies to evaluate minimal perturbation directly considering the output recommendations.

The pivotal research on the impact of adversarial attacks on image data in the case of recommender systems is justifiable by the fact that the computer vision field where the adversarial machine learning starts to get the interest from the research community. A few works focus on attacks on data other than images such as text, audio signal, and video. For instance, Carlini et al. [27] presented an adversarial attack strategy to craft targeted adversarial audio samples such that the transcription is entirely different while preserving the original soundtrack.

We may imagine an adversary modifying the audio signal of tracks in a music recommender system such that the text is associated with an explicit (illegal) content

or makes a love song (with an original romantic text) classified as a heavy metal track. Another example might be related to the injection of adversarial reviews in a review based recommender system. For instance, Gao et al. [47] proposed a black-box attack strategy to generate a spam movie review, which is classified as a positive message by the RNN-based classifier. This algorithm might drastically change the recommendations of a review-based recommender since spam review will alter the actual feedback from users. Following the same analysis, video adversarial samples [102] for video recommender, adversarial attack on graph-structured data [36] for social-based recommender, and adversarial input sequences for RNN [77] for sequence-aware recommendations are some intuitive examples that any hybrid/content.based recommender systems may be affected by adversarial attacks on their input data.

Defenses
This section defines the main principles to follow to protect a hybrid/content-based recommender system from the adversarial input samples. We define three main strategies: implement state-of-the-art (1) robustification procedure and (2) detection techniques of adversarial samples commonly used in the research field of used ML component to extract the content features, and (3) propose novel recommender models robust to adversarial samples.

Increasing the Robustness of the Feature Extractor The first category of defense mechanisms consist of selecting from the feature extractor research field (e.g., computer vision in the case of image classifier [1] and natural language processing for text classification [116]) the state-of-the-art approaches to make models more robust to adversarial samples. The main idea of robustification is to make the model able to be sensitive as little as possible to adversarial and real inputs data. Common strategies are:

- Model Robustification, e.g., Adversarial Training, and Defensive Distillation (see Sect. 3.1)
- Others, e.g., Deep Contractive Network [51], Data Compression [43], and Data Randomization [106]

It is essential to clarify to the practitioner that each defense strategy has to be evaluated based on the definition of the adversary threat model [24].

However, the application of robustification techniques of deep neural networks does not guarantee the recommender model's robustness. For instance, Anelli et al.[7] studied the efficacy of common defense strategies in the computer vision field, e.g., Adversarial Training and Free Adversarial Training, to the image feature extractor component used in state-of-the-art visual-based recommender systems (e.g., VBPR). After defining the adversary threat model, the authors demonstrated that the attacks still remain effective in manipulating the performance towards the adversary's malicious goal, leaving us an open challenge related to the study of defense mechanisms on the DNNs that effectively protect a recommender.

Detecting Adversarial Samples The second category of defensive mechanisms is composed of detection techniques. They aim to reject input data when classified as adversarial samples. The main detection techniques are:

- Classical ML Detector: PCA, Softmax, and Reconstruction of Adversarial Images;
- Adversarial Deep Detector: train a DNN to classify original/adversarial samples;
- Distributional Detection: filter out adversarial samples by comparing the distribution of the original images to the adversarial ones. The main methods are based on Kernel Density and Bayesian Uncertainty Estimation.

It is worth noticing that detection techniques have to be chosen based on the adversary threat model. For instance, Carlini and Wagner [26] verified the vulnerability of ten detection techniques under three adversary threat models. Furthermore, within this experimental setting, the authors have been able to demonstrate that the ten tested detection methods might be utterly useless in a scenario with a strong adversary (white-box attacks).

Increasing Robustness of the Recommender Models The last strategy to evaluate is identifying robustification methods to protect the model from the perturbability of content data. For instance, Tang et al. [91] modeled the adversarial perturbation directly applied to items' images with the perturbation of the features used in the visual-based recommender model. In this regard, the authors have used the adversarial training strategy presented in Sect. 2.4.2 by adversarially regularizing with respect to the image features. To the best of our knowledge, this is the first attempt to robustify a recommender model to limit the impact of input perturbation.

3.3 Machine-Learned Data Poisoning Attacks

Up to this section, the analysis fostered a broad class of attacks on machine learning models. An essential characteristic of those attacks is that they occur during the learning phase when the model is updated. However, another broad class of attacks targets the learning algorithms by manipulating the data used for training these models. The practitioners generally refer to these kinds of attacks as Poisoning attacks. For the sake of clarity, it is beneficial to consider that there are several categories of poisoning attacks. These categories are mainly related to the kinds of modifications on training data the adversary can perform. Moreover, some poisoning attack models will typically either impose a constraint on the number of modifications or a modification penalty; they may also constrain what the attacker can modify about the data (e.g., feature vectors and labels, only feature vectors or only labels), and what kinds of modifications are admissible (e.g., insertion only, or arbitrary modification).

The first kind of attack is named **Label modification attack**: this attack lets the attacker alter some supervised learning labels (within a specific overall budget).

Then, there is the **Poison insertion attack**, in which the attacker can attach some new poisoned feature vectors to the original data, with (or without) the corresponding target label. Another common kind of attack is the **Data modification attack**, where the attacker can modify some existing feature vectors or the labels (or both) in the original data before the model training. Finally, in **Boiling frog attacks**, the attacker exploits the frequent retrainings to poison the model at each retraining imperceptibly. In the literature, several methods have been proposed to perform Poisoning Attacks. Most of them consist of empirical techniques developed to conduct the attacks. Nevertheless, since these algorithms do not provide for any optimization (or adversarial learning), they remain partially outside the section's scope. Here, we drive the reader's attention to all those techniques that not only propose a novel Poisoning Attack but also propose *a specific optimization procedure to maximize the adversary's goal automatically.* To this extent, the underlying data model is crucial to define a proper optimization criterion. For this reason, we first introduce the methods that involve factorization models since this recommendation model is widely adopted for recommender systems, and the first attempts to realize the Poisoning Attack Optimization have been realized by exploiting this model.

3.3.1 Data Poisoning Optimization on Factorization-Based Recommenders

This section presents a systematic review of techniques for computing near-optimal data poisoning attacks for factorization-based recommendation models. In general, all these works share a factorization-based recommendation model, consisting of a very sparse URM matrix and the factorized vectors representing the users and the items. Despite the commonalities related to the underlying primary model and some mathematical choices for the gradient computation, the approaches we present in this section differ in other aspects. For a matrix consisting of m users and n items, the attacker is capable of adding a small fraction $\alpha \cdot m$ of malicious users to the training data matrix, and each malicious user is allowed to rate at most B items with each preference bounded in the recommender specific range $[-\Lambda, \Lambda]$.

The first relevant work was proposed by Li et al. [64] in 2016. Their intuition was to exploit the techniques for optimizing malicious data-driven attacks proposed in previous literature regarding machine learning algorithms' security. Since poisoning attack model optimization can be modeled as a constrained optimization problem, some previous works in the ML field focus on how to approximately compute implicit gradients of the solution of an optimization problem based on *first-order KKT conditions.* Li et al. exploit this mathematical background and define a new unified optimization framework for recommender systems for computing optimal attack strategies. In this pioneering work, the authors exploit the *Projected Gradient Ascent (PGA)* method for solving the problem of maximizing the attacker utility. They characterize several attacker utilities, including **availability attacks**, where the goal is to increase the prediction error, and **integrity attacks**, where item-specific

objectives are considered. They assume an attacker with knowledge of both the learner's learning algorithms and parameters (that follow the *Kerckhoffs' principle* to ensure reliable vulnerability analysis in the worst case). Finally, they provide the optimization techniques, based on *first-order Karush-Kuhn-Tucker (KKT) conditions*, for two popular minimization algorithms: alternating minimization and nuclear norm minimization.

Data poisoning optimization was further investigated for factorization-based social recommendation models. In Hu et al. [57], the authors have considered two different types of attack actions for the attacker: **(1)** injecting fake ratings and **(2)** generating fake relationships with normal users. In the loss function, the model considers an additional factor that exploits the cosine similarity between the user's friends and the target items. In the targeted poisoning attack, the attacker aims to promote an item to target users by injecting fake ratings and edges. We assume the recommender system recommends a list of items to the target user, consisting of N unrated items with the largest predicted rating scores. If the item is in the list, the attacker has successfully attacked the target user. Therefore, to attack the target user, the attacker's goal is to maximize the hit probability of finding the target item in the recommendation list. Since it is hard to model the hit probability directly, analogously to prior works [44], they adopt the *Wilcoxon-Mann-Whitney loss* [12] to approximate it. However, the final goal is to model the attack as an optimization problem. Let the attacker control a set of malicious nodes. Hence, the attacker aims to find the optimal rating score vector and edge weight vector for all malicious nodes to minimize the total attack loss. The authors formulate the targeted poisoning attack as a *bi-level* optimization problem that considers the attack loss for each target user.

For the sake of completeness, several works in this section employ an approximation of the adversarial gradient. However, Tang et al. [92] have recently proposed a method that exploits the same adversarial gradient.

Furthermore, to avoid simple malicious node detection methods based on the number of ratings, the authors assume that each malicious node can rate *at most* a certain number of items. While for bi-level optimization problems with continuous variables, the literature proposes approximate solutions by exploiting gradient descent methods based on the lower-level problem *KKT conditions*, such methods cannot be directly applied to this problem since it involves discrete variables. Therefore, they propose a framework to approximately solve the problem. Indeed, the authors avoid optimizing the fake data (i.e., the rating scores and edge weights) of all malicious nodes simultaneously. Instead, they propose to optimize the fake data sequentially. Given the original rating scores and edge weights, as well as the fake rating scores and edge weights that were added so far, they find the rating scores and edge weights for the next malicious node to minimize the attacker's loss. Then, they alternatively generate fake rating scores with fixed fake edge weights for each malicious node and then generate fake edge weights with fixed fake rating scores until convergence.

Another critical and complex recommendation scenario is Cross-Domain recommendation. Even more here, where the recommendation takes place by leveraging information in two different domains, personal data security, and privacy are crucial

investigation fields. In this sense, Chen et al.[31] have proposed a data poisoning attack framework to conduct several kinds of attacks. Again here, according to the *Kerckhoffs' principle*, they have considered the worst-case scenario in which the attackers have full knowledge of the attacked learning algorithm, namely *perfect knowledge attacks*. In detail, the attacker can inject a *certain number* of malicious users in the source domain and observe the effects on the learning algorithm and the model in the target domain. They rely on a particular factorization model, the *Collective Matrix Factorization*, that is a popular approach for dealing with pairwise relational data. Specifically, the two factorized models (in the source and the target domain) are computed simultaneously by exploiting a joint loss function. Hence, they formulate the optimal attack problem as a *bi-level* optimization problem. The outer optimization maximizes the attackers' utilities, while the inner optimization maximizes the poisoned data models' recommendation utility. In particular, the authors focus on three kinds of data poisoning attacks:

- **Availability attack**: Attackers try to maximize the estimated error in the target domain. The utility function is here formulated as the total amount of perturbation of attacked items scores on unobserved elements.
- **Integrity attack:** Attackers' goal is to increase (or decrease) the popularity of a subset of items.
- **Hybrid attack:** A mixture of availability attack and integrity attack.

The trade-off factors between the availability and the integrity attack in the utility function provide remarkable freedom. Indeed, the factors can be negative whether the attackers want to increase their items' popularity while perturbing less other recommendations to avoid detection.

They then transform the bi-level optimization problem to a single-level constrained optimization problem by exploiting the KKT conditions.

The adoption of learning-to-rank optimization criteria in Recommender Systems most certainly represents the major advance in the last decade. In 2020, Fang et al. [45] have proposed a relevant work that tackles the data poisoning optimization for the top-N recommendation. Here, the attacker's goal is to promote a particular item to as many normal users as possible and maximize the hit ratio, which is defined as the fraction of normal users whose top-N recommendation lists include the target item. The assumption is that the attacker is able to inject *some* fake users into the recommender system. Moreover, as in the other works, the attacker has *full knowledge* of the target recommender system (e.g., all the rating data, the recommendation algorithm). This work introduces several novelties and ideas.

- On the one hand, they propose to use a loss function to approximate the number of users to whom the target item will be recommended. They adopt a standard solution to solve the discrete rating problem: relaxing the integer rating scores to continuous variables and convert them back to integer rating scores after solving the reformulated optimization problem.

- On the other hand, they exploit the influence function approach inspired by the interpretable machine learning literature that shows that the top-N recommendations are mainly affected by a subset S of influential users.

For this reason, they refer to the proposed attack as **S-attack**. Finally, given S, they propose a gradient-based optimization algorithm to determine the fake users' rating scores. In detail, they optimize the rating scores for fake users *sequentially* instead of optimizing for all the fake users simultaneously. In particular, they optimize the rating scores of a single fake user and add the fake user to the recommender system. Even here, the *hit ratio maximization problem* (*HRM* problem) is challenging, even if only one fake user is considered. To address the problem, they use a differentiable loss function to approximate the hit ratio (Wilcoxon-Mann-Whitney loss). However, the most important novelty is that, instead of using all regular users, they only use a selected subset of influential users to solve the HRM problem. Finally, they develop a gradient-based method to solve the HRM problem to determine the fake user's rating scores. Indeed, it has been observed [62, 101] that different training samples have different contributions to the solution quality of an optimization problem. The authors also propose a "standard" technique to detect fake users. Specifically, they extract six features: *RDMA* [33], *WDMA* [73], *WDA* [73], *TMF* [73], *FMTD* [73], and *MeanVar* [73] for each user from its ratings. Then, they build a training dataset consisting of an equal number of fake users generated by the attack and randomly sample regular users for each attack. Afterward, the training dataset is used to train a classifier for fake users' detection (here, a *Support Vector Machine*-based classifier).

The work presented in [56] is a novel **defensive strategy** that leverages *trim learning* to make matrix factorization resistant to data poisoning. *Trim learning* is a learning method that is robust against the data poisoning attack. Indeed, *trim learning* exploits the statistical difference between normal users and fake users as well as the differences between normal and fake items to learn a model while excluding the malicious information. Hence, the authors have integrated this learning method in the learning algorithm for matrix factorization, named *trim matrix factorization algorithm*. Finally, they have tested the defensive strategy's efficacy against some popular kinds of attacks: *Random*, *Average*, *Bandwagon*, and *Obfuscated Attacks*.

Differently from the previous works, Liu et al. [67] develop a **robustness analyzer** for factorization machines. Given a trained Factorization Machine (FM), and a perturbation space, the goal is to provide a robustness certificate which states whether the FM's prediction is robust or non-robust against data poisoning attacks. Once an instance is certifiably robust, its label does not change irrespectively to the attack models exploited in the given perturbation space. They decided to focus their work on factorization machines (FMs) thanks to their ability to handle discrete and categorical features. A common solution to deal with these features is to convert them to binary features via one-hot encoding or multi-hot encoding. Since the number of possible values is large, the resulting discrete feature vector can be high-dimensional and sparse. FMs incorporate the interactions between features building

an accurate recommendation model. In detail, the FM prediction formula considers the combination of multiple individual features (i.e., it creates a plethora of derived features). The problem arises when an attacker *slightly modifies the input data*, and very similar instances receive very different predictions. In this paper, the authors model the recommendation problem as a binary classification. Therefore, attacked instances might be classified into completely different classes. These possible unreliable results can significantly limit the applicability of Factorization-based recommenders. In this respect, the authors sensed the importance of investigating adversarial perturbations' effects on the factorization machines. To certificate the robustness of an FM, they derive the *upper/lower bounds* that can be reached by all the possible perturbations (with a certain budget). If the bound is *certifiably robust*, no perturbation can change the prediction of the instance.

3.3.2 Data Poisoning Optimization and Reinforcement Learning

In contrast to the previous section, the attacks based on Reinforcement Learning (RL) show several similarities. Indeed, irrespective of the specific learning algorithm, the description of the method here requires to describe several aspects:

- the **attacker's knowledge** and **capability** (that are common also to the other attack families);
- the **state space**;
- the **action space**;
- the **reward utility**.

Beyond these commonalities, it is worth noting that Reinforcement Learning is a broad research field, and the specific techniques that are adopted are typically different depending on the scenario, the available information, and the capabilities of the attacker to act in the environment. Here we focus on two attack frameworks, **LOKI** [115] and **PoisonRec** [88].

The reinforcement learning-based framework **LOKI** learns an attack agent to generate adversarial user behavior sequences for data poisoning attacks. Differently from the other attack methods that we have previously analyzed, reinforcement learning algorithms *leverage the feedback from the recommendation systems*, instead of the whole architecture and parameters, *to learn the agent's policy*. The first limitation of this approach is that the attacker hardly controls the target recommendation system to be retrained to get the feedback and update the attack strategy. Moreover, recommendation services generally limit feedback frequency. Unfortunately, a reinforcement learning-based framework requires a large amount of feedback to train a policy function efficiently. For these reasons, the authors decided to circumvent the exploitation of real feedback from the target recommendation system to train a policy. In detail, they build *a local recommender simulator* to mimic the target model and make the reinforcement framework get reward feedback from the recommender simulator instead of the target recommendation system. The recommender simulator is built as an ensemble of *several representative*

recommendation models to be agnostic to the real recommender system. This choice is based on the assumption that if two recommenders can produce similar recommendation results on a given dataset, then the adversarial samples generated for one of the recommenders could be used to attack the other. However, even though they rely on a local recommender, the number of retrainings needed for a reinforcement learning framework makes the retraining impracticable. Therefore, the authors develop an *outcome estimator* based on the influence function. The outcome estimator estimates the influence of the injected adversarial samples on the attack outcomes.

Let us summarize the main aspects of the scenario that the authors analyze, and that makes the current scenario differ from the previous ones:

1. First, they suppose a specific recommendation scenario, the *next-item recommendation task*; In this setting, given the existing sequences, the next-item recommendation's goal is to produce an ordered item list, which predicts the next items that the user will choose;
2. The attacker's goal is to promote a set of target items to as many target users as possible;
3. The attackers can inject fake users into the recommendation system. These users can visit or rate items, but these actions are limited to make the profile unnoticeable;
4. The attacker can access the full activity history of all the users in the recommendation system. The critical difference with a classic attack is that this knowledge is not known a priori;
5. As in previous works, even here, the number of fake users is limited;
6. Finally, the attacker does not know when the recommendation model is retrained and can only receive a limited amount of feedback from the black box recommendation model.

In the setting depicted by the authors, the data poisoning problem can be formulated as the creation of new sequential patterns that involve the target items in the training set of the target recommendation system. Consequently, the adversarial samples' generation is a *multi-step decision process*, in which the generator ought to select specific actions for the controlled users to *maximize attack outcome*. From the perspective of reinforcement learning, the *goal is to learn a policy function to generate sequential adversarial user behavior samples*, which can maximize the target users' averaged display rate. Hence, the reinforcement learning-based framework **LOKI** consists of a *recommender simulator*, an *outcome estimator*, and an *adversarial sample generator*. The adversarial attack against a local recommender simulator is essentially interpreted as a *multi-step decision problem*. This means that the attack process should be modeled as a *Markov Decision Process* (MDP). Typically, an MDP is defined by a ***set of states***, a ***set of actions***, the ***transition probabilities***, and the ***rewards***. In the considered scenario, the **Action space** consists of the possible actions that the agent can act. Since considering every action strategy for each item is particularly expensive and inefficient, the authors propose to divide the item set

into groups and use the set of all the groups as action space. Indeed, they show that *adversarial samples do not necessarily need to match the same sample pattern*.

The **state** is defined as the sequence of actions that precede the current step. Finally, since the RL framework should learn a policy that increases the estimated prediction scores of target items, the authors propose a **reward function** that considers the *predicted weighted average influence of the target samples*. Additionally, **actions** are represented by means of an *embedding layer*.

In contrast with the two-fold motivation behind LOKI, the authors of **PoisonRec** focus on the myriad of different recommender algorithms currently available. In their opinion, indeed, it would be *"very difficult and also time-consuming to design effective attacks for each of them directly"*. To learn effective attack strategies on different complex black-box recommender systems, they resort to *Model-free reinforcement learning*, which requires just a little knowledge of the recommender system. From this perspective, they pose only *light constraints* to their recommendation scenario:

- In fact, the authors suppose a limited knowledge about the recommender system that could even be a black-box;
- They assume to know nothing about the logs, the system components, and the recommendation algorithm.
- Beyond this, the attackers can only retrieve basic item information like item title, item description, and an item's sales volume.

PoisonRec repeatedly injects fake user actions into a recommender system while improving the attack strategy by exploiting the rewards. In detail, as observed with LOKI, the authors of PoisonRec model the sequential attack behavior trajectory as an MDP. However, the choice of the **reward function** is not trivial at all. They take advantage of a widely-adopted metric, the number of *Page View* (PV), that measures items' exposure within a certain period in the online recommender system. Indeed, the attacker's goal will be *maximizing the Page Views* for certain target items. Although **PoisonRec** can not obtain PV for other users when attacking a real recommender system due to the black-box setting, PV is a usually available statistical indicator. Indeed, the attackers know how many users have viewed their products and other platform-specific indicators. In **PoisonRec**, the *attack trajectory* is defined as an MDP, with the additional constraint that *all the attackers will share the same policy network*. In this MDP, the **state space** consists of the current attacker and all the selected sequential actions performed so far. The **action space** is the union between the items in the catalog and the target items. Finally, as mentioned earlier, the **reward function** corresponds to the target items' PVs. In the policy network, the authors exploit two neural networks, an *LSTM* network and a *deep neural network* (DNN).

3.3.3 Data Poisoning Optimization Poisoning with Other Recommendation Families

Although the factorization-based and the reinforcement learning-based data poisoning methods have aroused much research interest in the last years, even other recommendation families deserve to be in the spotlight.

For example, **graph-based recommender systems** are becoming increasingly popular in the last decade. The reasons for their success are manifold.

On the one hand, the graphs let designers develop a *complex recommender system that considers heterogeneous classes*. Indeed, it is possible to build a *multi-partite graph* in which each partition contains *ontologically* different entities.

Second, we have witnessed a flourishing of advanced techniques to explore, summarize, and embed graphs. These techniques are now a swiss knife in the hands of the recommender systems practitioner.

In 2017, Yang et al. [108] proposed injecting fake co-visitations into the system. They supposed a bounded number of fake co-visitations that an attacker could inject, focusing on the items (and the number of fake co-visitations) to inject. By exploiting a strategy similar to the previous approaches, they modeled the attack as a constrained linear optimization problem to perform attacks with maximal threats.

Later, Fang et al. [44] have considered a graph-based recommender system in which the recommendation is realized by exploiting the stationary probabilities generated by random walks. As in the other works, they suppose the presence of an attacker who wants to *promote/demote a set of items*. Even here, they need to define a differentiable loss that might drive the learning. The choice falls on one of the most popular recommendation metrics, the *Hit Ratio*, which could be used to indicate the attack's efficacy. However, the *Hit Ratio* itself is not differentiable. Thus, the authors decide to rely on a proxy function, the *Wilcoxon-Mann-Whitney loss*, known to optimize the ranking performance. Once the model and the optimization function is defined, the authors focus on solving the optimization problem. The resulting optimization procedures *updates the weights of the edges* (the probabilities) *iteratively*. At the end of the learning process, the higher weights will represent the near-optimal path to the items that will compose the filler item set.

The authors also develop a method for detecting fake users inspired by *Sybil detection* in social networks. In detail, they propose a behavior-based method. First, they extract a set of features from a user's rating scores. Then, they train a classifier to distinguish between normal and fake users. In this respect, they have considered the features proposed in the previous literature: *Rating Deviation from Mean Agreement* (RDMA) [33], *Weighted Degree of Agreement* (WDA) [73], *Weighted Deviation from Mean Agreement* (WDMA) [73], *Mean-Variance* (MeanVar) [73], *Filler Mean Target Difference* (FMTD) [73].

Neighborhood-based recommendation algorithms is the main topic of [32]. The rationale behind this work is to push a set of items by taking advantage of the neighborhood mechanism. While there are several poisoning attacks to k-NN methods, to date, this represents the first attempt to *learn an optimal set of fake users*

automatically for k-NNs through a data poisoning attack optimization problem. Indeed, Chen et al. [32] first define the data poisoning attack as an optimization problem, and then approximate the optimization problem to generate fake users. The authors introduce realistic constraints to limit the number of fake users and the maximum number of filler items. The authors then propose three different loss functions to approximate the *Hit Ratio*: **(1)** based on similarities, **(2)** based on user ratings, **(3)** a combination of both.

4 Evaluation

In this section, we discuss and analyze the methodologies to evaluate adversarial recommender systems. First, we present the experimental settings commonly adopted in the literature. Then, we introduce the evaluation metrics for evaluating the three-fold adversary's goals.

4.1 The Experimental Setting

Beyond the classical recommendation evaluation protocols , we need to consider a few other evaluation dimensions in Adversarial Learning . Due to the specific experimental scenarios, it is crucial to measure the recommendation's performance both in a *clean* and in an *attacked* setting. For this reason, a first evaluation takes place before running the attack or the defense. Thereby, we can evaluate the recommender's overall performance to assess the *a posteriori* impact of the system under evaluation. Since adversarial learning can support the either attack and defensive strategies, the evaluation settings can be categorized as:

- **Evaluation of Attack Strategies.** It concerns the comparison of the efficacy of adversarial attack methods **A** against a recommendation system **R**. It is common to measure it through the variation of performance before and after the execution of **A**. In this evaluation, it is necessary to check that the attacks have the same adversary threat models, i.e., the knowledge and the capabilities. For instance, comparing a *Zero-Knowledge* with a *Perfect-Knowledge* adversary is meaningless since typically, the more knowledge an attacker has, the more powerful attacks are.
- **Evaluation of a Defense Strategy.**
  - *Evaluation of the Robustification.* It compares the effectiveness of a defense strategy **D** against an adversarial attack method **A** on a recommender system **R**. Here, the evaluation goal is to evaluate the impact of **A** against both **R** and $\mathbf{R} + \mathbf{D}$ (i.e., the defended recommender system). It typically entails evaluating the variation of the performance in the clean setting of **R** and $\mathbf{R} + \mathbf{D}$. The

variation gives an idea of the influence of the defense strategy in the clean setting.
– *Evaluation of Detection Techniques.* To evaluate the detection of malicious input data (e.g., images, audio tracks, fake profiles), it is common to adopt accuracy metrics like *precision* and *recall*. Moreover, even where the method correctly removes the malicious data, it is crucial to investigate the effects of not detecting the malicious sample (if the techniques were not applied).

4.2 Evaluation Metrics

The comparison of attack algorithms has its roots in the adversary's goal. As previously introduced in Sect. 3, the adversarial attack strategies may differ profoundly depending on the domain, the recommendation task, and the aspects the attacker decides to target. One adversary may want to reduce the overall recommendation performance or influence the item recommendation task to push/nuke a product or a set of products. This kind of attack can be evaluated by exploiting protocols that consider usual recommendation metrics. These protocols are not sufficient to cover all the cases. For instance, an adversary may attack the content data (e.g., images, music tracks, reviews) to influence the automatically recognized classes that categorize the item. It is needed to resort to additional metrics to measure the human-perceptibility or other aspects of the adversarial variation in such cases. Even though it is impossible to exhaust all the possible evaluation scenarios, the following sections are a broad overview of the multiple scenarios we can deal with in an adversarial learning setting.

4.2.1 Impact on Overall Recommendation Performance

The final goal of these attack strategies is to alter the system's recommendation performance. For the sake of simplicity (and even because this is the generally adopted perspective), here we consider the situation in which the attacker aims to reduce the recommendation effectiveness. In the literature, the worsening of recommendation performance is evaluated with the percentage variation of two classical rank-wise accuracy measures for the item recommendation task: *Hit Ratio* ($HR@k$) and *normalized Discounted Cumulative Gain* ($nDCG@k$). Let $HR_{\mathrm{i}}@k$ be the hit ratio measured in the clean setting, and let $HR_{\mathrm{f}}@k@k$ be the same metric evaluated after the application of an adversarial attack. The percentage variation of the hit ratio ($d_{HR@k}$) is defined as follows

$$d_{HR@k} = \frac{HR_{\mathrm{f}}@k - HR_{\mathrm{i}}@k}{HR_{\mathrm{i}}@k} \times 100 \tag{26}$$

Analogously, the percentage variation of the normalized Discounted Cumulative Gain ($d_{nDCG@k}$) is defined as:

$$d_{nDCG@k} = \frac{nDCG_{\mathrm{f}@k} - nDCG_{\mathrm{i}}@k}{nDCG_{\mathrm{i}}@k} \times 100 \tag{27}$$

For both metrics, the more negative values there are, the more impactful the attack is. In fact, these two metrics serve to study both adversarial attacks on model parameters [30, 55] and content data [91].

Since the adversary may also aim to alter other recommendation aspects, it is possible to apply the percentage variation measures to any accuracy and beyond accuracy metrics. Anelli et al. [6] evaluated the impact of iterative adversarial perturbations by analyzing a broad spectrum of metrics. There, the analysis covers item coverage [48], Shannon Entropy [94], and expected free discovery [97], as well as precision and recall accuracy measures.

4.2.2 Impact on the Recommendability of Item Categories

The second type of evaluation has a much specific focus. It considers the performance variation when the adversary targets to push or nuke an item or a set of items. It is worth noticing that this evaluation is different from evaluating the effect of shilling attacks. The item or segment of items is selected in those attacks without considering any relation to the items' content. On the other hand, the evaluation of these attacks considers that content data can be perturbed. The injected perturbation tries to fool the classifier used to extract the high-level features used in a hybrid/content-based recommendation system. For instance, in a real scenario, the attacker may adversarially perturb source product images such that the image feature extractor component will misclassify them. The new classification may make the item fall into a different category that is more popular or less popular compared to the source products. While under a general machine learning perspective, it would be essential to evaluate the attacker's capability to move the item into another class, in a recommendation scenario, the core aspect is the recommendation performance variation of the classes involved in the attack. Analogously, a similar evaluation can occur in a music recommender system when the adversary perturbs soundtracks, where the deep neural classifiers will misclassify them with different song genres (e.g., heavy metal instead of pop). Following this line, a plethora of metrics could be proposed to assess the category-specific recommendation performance. Here, for the sake of space, the focus remains on the two most adopted metrics in adversarial literature on recommendation systems: *Hit Ratio* and *nDCG*.

The **Category Hit Ratio** (CHR@k) [75] is a variant of HR that evaluates the fraction of adversarially perturbed items in the top-K recommendations.

Definition 8 (Category Hit Ratio) Let $\mathscr{C}$ be the set of classes for a classifier, $\mathscr{I}_c = \{i \in \mathscr{I} \mid x_i$ is classified as $c \in \mathscr{C}\}$ be the set of items whose content information x_i

is classified as c. The categorical hit (*chit*) is defined as:

$$chit(u,k) = \begin{cases} 1, & \text{if } k\text{-th item} \in \mathscr{I}_c \\ 0, & \text{if } k\text{-th item} \notin \mathscr{I}_c \end{cases} \tag{28}$$

Analogously, the $CHR_u@K$ is defined as:

$$CHR_u@K = \frac{1}{K}\sum_{k=1}^{K} chit(u,k) \tag{29}$$

Categorical hit ($chit(u,k)$) is a 0/1-valued function that returns 1 when the item in the k-th position of the top-K recommendation list of the user u is in the set of attacked items not-interacted by u. Since *Category Hit Ratio* does not consider the ranking (and relevance) of recommended items, **Category normalized Discounted Cumulative Gain** has been proposed [7], that assigns a gain factor to each considered ranking position. By considering a relevance threshold τ, each item $i \in \mathscr{I}_c$ has an ideal relevance value of:

$$idealrel(i) = 2^{(s_{max}-\tau+1)} - 1 \tag{30}$$

where s_{max} is the maximum possible score for the items in the recommended list. By considering a recommendation list provided to the user u, the relevance $rel(\cdot)$ of a suggested item i is defined as:

$$rel(k) = \begin{cases} 2^{(s_{ui}-\tau+1)} - 1, & \text{if } k\text{-th item} \in \mathscr{I}_c \\ 0, & \text{if } k\text{-th item} \notin \mathscr{I}_c \end{cases} \tag{31}$$

where k is the position of the item i in the recommendation list. In Information Retrieval, the *Discounted Cumulative Gain* (DCG) is a metric of ranking quality that measures the usefulness of a document based on its relevance and its position in the result list. Analogously, Category Discounted Cumulative Gain ($CDCG$) is:

$$CDCG_u@K = \sum_{k=1}^{K} \frac{rel(k)}{\log_2(1+k)} \tag{32}$$

Since recommendation results may vary in length depending on the user, it is not possible to compare performance among different users, so the cumulative gain at each position should be normalized across users. In this respect, an *Ideal Category Discounted Cumulative Gain* ($ICDCG@K$) is defined as follows:

$$ICDCG@K = \sum_{k=1}^{min(K,|\mathscr{I}_c|)} \frac{rel(k)}{\log_2(1+k)} \tag{33}$$

$ICDCG@N$ indicates the score obtained by an ideal recommendation list that contains only relevant items. We can finally introduce the *normalized Category Discounted Cumulative Gain* ($nCDCG$) defined as:

$$nCDCG_u@K = \frac{1}{ICDCG@K} \sum_{k=1}^{K} \frac{rel(k)}{\log_2(1+k)} \tag{34}$$

$nCDCG_u@K$ is ranged in the [0, 1] interval, where values close to 1 mean that the attacked items are recommended in higher positions (e.g., the attack is adequate).

4.2.3 Qualitative Evaluation of Perturbed Content

In this section, we present qualitative measures for assessing the perceptibility of adversarially perturbed content data in multimedia recommenders. For instance, accuracy-oriented evaluation metrics are insufficient to evaluate the scenario of attacking a recommendation system by perturbing content data. Indeed, an adversary might prepare a powerful attack that effectively alters the recommendation performance. However, the degradation of the perturbed content would be too evident. For instance, the final user would become aware of the attack because the image of a recommended product is altered or the audio track of a song is noisy. For this reason, qualitative perceptual metrics can evaluate the user perceptibility of the content perturbations. Moreover, these metrics can help to design an effective detector for maliciously manipulated content. In the following, we introduce a set of commonly adopted metrics with three distinct content categories: images, audio, and text.

Images For evaluating the quality of image perturbations, it is crucial to assess if users can detect a manipulated image. In an offline evaluation setting, the following metrics are often adopted:

- *Distance Metrics.* [111] The norm distances are widely adopted in the computer vision community to assess the perceptibility of perturbations. Based on the norm value p, these metrics are usually identified as l_p-norm distances.
- *Peak Signal-To-Noise Ratio* (PSNR) [103] is a more easily interpretable, logarithmic version of the Mean Squared Error (MSE).
- *Structural Similarity Index* (SSIM) [100] is a metric based on the assumption that humans are sensitive to the image's *structure*.
- *Learned Perceptual Image Patch Similarity* ($LPIPS$) [114] produces a perceptual distance value between two similar images by leveraging on (1) knowledge extracted from convolutional layers inside state-of-the-art CNNs and (2) collected human visual judgments about pairs of similar images.

Audio The metrics measure if the adversarial perturbation has exceeded the hearing thresholds. Unfortunately, the signal-to-noise ratio (SNR) is insufficient to

determine the amount of perceptible noise since it does not represent a subjective metric. The main offline evaluation metrics are:

- *Distortion Metric.* [74] It quantifies the distortion introduced by adversarial perturbation by exploiting a l_∞ distance metric.
- *Psychoacoustic hearing thresholds* [83] calculates the differences between the original and the modified signal spectrum to the threshold of human perception.
- *Perceptual Evaluation of Speech Quality (PESQ)* [81] integrates the perceptual analysis measurement system with the perceptual speech quality measure.

Text For text-based content data (e.g., reviews given by users, product descriptions, tweets, or social posts), the evaluation is usually conducted with specific variants of l_p-norm distance measures. Indeed, small changes in a text (e.g., words, characters) are more readily perceptible. These evaluation metrics [116] have to measure the minimum perturbation amount to fool the system while remaining human-unperceivable (the lexical, syntactic, and semantic correctness must be preserved even with small distance-based measures). Among them, we cite: *Norm-based measurement*, *Grammar and syntax related measurement*, *Semantic-preserving measurement*, *Edit-based measurement*, *Jaccard similarity coefficient*.

5 Conclusion and the Road Ahead

Combined with the growing abundance of large-high-quality datasets, substantial technical breakthroughs over the last few decades have made machine learning (ML) a vital tool across a broad range of tasks, including computer vision and natural language processing and recommender systems (RS). However, success has been accompanied by a significant new arising challenge: *"many ML applications are adversarial in nature"* [98].

Modern RS utilize latent factor models (LFMs) such as matrix factorization (MF) as the core predictor and optimize it with pairwise ranking objectives, such as the Bayesian personalized ranking (BPR). This combination of LFM+BPR optimization is the most prominent adopted strategy to date to drive item recommendation tasks in different settings and domains. However, despite their great success, it has been shown that this combination leads to recommender models that are not *robust* in the face of adversarial attacks, i.e., subtle but non-random perturbation of model parameters that are found via (solving) an attack strategy, leading to the resultant model to produce erroneous predictions. Therefore, to enhance the robustness of recommender models, a new optimization framework has been designed, named adversarial personalized ranking (APR) [55], where it has been shown that it can not only perform more robust against adversarial samples but also produce a better generalization performance on clean samples.

Throughout this book chapter, we have reviewed the current state of the art in adversarial attacks against recommendation models. Adversarial machine learning for recommender systems (AML-RecSys) [3] combines best practices in ML

and security to improve data security in RS tasks. We provided a taxonomy of adversarial RS that classifies AML-RecSys according to a novel classification point, level of granularity, according to (i) adversarial perturbation of model parameters, (ii) adversarial perturbation of content and, (iii) ML-optimized data poisoning attacks in which (iii) is an additional contribution we offer in this work to give a complete picture of AML for the security of RS. This novel classification axis provides a pragmatic approach to implement adversarial attacks against a specific given recommendation model.

Despite the many new and challenging results reached in the field we have seen in this chapter, there are still open issues and research directions that need further investigation [41]. Among them, we can indeed mention the following open challenges.

Find New Attack/Defense Models in the RS Domain to Better Exploit the Theoretical and Practical Results Obtained in the ML/CV Field First, we need to understand what is for RS the equivalent notion of *human-imperceptible or inconspicuous* we find in CV. As a second aspect, images are represented as continuous-valued data, while in RS, we deal with discrete data representing user profiles.

Assess the Effects of Adversarial Attacks on Item/User Side Information Most of the proposed strategies mainly deal with the collaborative data available in the user-item matrix. Nevertheless, modern recommendation models exploit a wealth of side-information beyond the user-item matrix, such as multimedia content presented in Chapter "Multimedia Recommender Systems: Algorithms and Challenges", social-connections, semantic data, among others [86]. Investigating the impact of adversarial attacks against these heterogeneous data remains an interesting open challenge.

Definition of the Attack Threat Model The research in the RS community misses a common evaluation approach for attacking/defending scenarios such as the one introduced by Carlini at el. [24]. For instance, it is essential to define a standard threat model to establish the attacker's knowledge and capabilities to make the attack (or defense) reproducible and comparable with novel proposals.

Model Stealing Another relevant area to the security of RS concerns the idea of model stealing in which the attacker can query a classifier, e.g., a web-based RS, and use the classifications labels (i.e., recommendation results) for reverse-engineering the model. While in adversarial attacks and data poisoning attacks, which we presented in this book chapter, the adversary typically has some knowledge about the structure of the model and tries to learn the model parameters, in model-stealing attacks, these assumptions are relaxed. The goal of these systems is to build a functionally equivalent classifier without knowing about the model type, structure, and parameters [87].

Effects on Beyond Accuracy Metrics Most of the research on adversarial ML and RS focuses on accuracy metrics, e.g., HR and nDCG. The impact on beyond

accuracy metrics could be, in principle, the main objective of a new breed of attack strategies aiming to compromise the diversity/novelty of results.

Scalability and Stability of Learning We identify the need to further explore the stability learning problems in the discrete item sampling strategy to train the generator of a GAN architecture. This has already been identified as a big problem when GAN-based recommenders are applied in real scenarios with huge catalogs. A point of study may be that of novel GAN models proposed in computer vision (e.g., WGAN [11], LSGAN [69], and BEGAN [14]).

Attacks on User Privacy and Fairness of RS and Protection Against It Another investigation area is related to protecting user privacy, in the face of growing attention to user ownership of data after major data impeaches such as Cambridge Analytica [49] and GDPR proposed by the European Union. Recently attempts have been made to build machine-learned recommendation models that offer a privacy-by-design architecture, such as federated learning [8, 9], or the ones based on differential privacy. Additionally, in the light of adversarial setting, more attention has been made to build a privacy-preserving framework that can protect users from adversaries that aim to infer, or reconstruct, their historical interactions and social connections, consider for example [71, 72]. Fair and unbiased recommendations [40] are also similar related concerns of users—and item providers—in the RecSys community, see Chapter "Fairness in Recommender Systems". To address these concerns, recent attempts are emerging that try to use AML to provide an unbiased ranking list [119] or build fairness-aware systems for specific domains such as news [105].

References

1. N. Akhtar, A. Mian, Threat of adversarial attacks on deep learning in computer vision: a survey. IEEE Access **6**, 14410–14430 (2018)
2. M. Aktukmak, Y. Yilmaz, I. Uysal, Quick and accurate attack detection in recommender systems through user attributes, in *RecSys* (ACM, New York, 2019), pp. 348–352
3. V.W. Anelli, Y. Deldjoo, T. Di Noia, F.A. Merra, Adversarial learning for recommendation: Applications for security and generative tasks - concept to code, in *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22–26, 2020* (ACM, New York, 2020), pp. 738–741
4. V.W. Anelli, Y. Deldjoo, T. Di Noia, E.D. Sciascio, F.A. Merra, Sasha: Semantic-aware shilling attacks on recommender systems exploiting knowledge graphs, in *The Semantic Web - 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31–June 4, 2020, Proceedings* (2020), pp. 307–323
5. V.W. Anelli, T. Di Noia, D. Malitesta, F.A. Merra, Assessing perceptual and recommendation mutation of adversarially-poisoned visual recommenders (short paper), in *DP@AI*IACEUR Workshop Proceedings*, vol. 2776, CEUR-WS.org (2020), pp. 49–56
6. V.W. Anelli, A. Bellogín, Y. Deldjoo, T. Di Noia, F.A. Merra, Msap: Multi-step adversarial perturbations on recommender systems embeddings, in *The International FLAIRS Conference Proceedings (FLAIRS 2021)*, vol. 34 (2021)

7. V.W. Anelli, Y. Deldjoo, T. Di Noia, D. Malitesta, F.A. Merra, A study of defensive methods to protect visual recommendation against adversarial manipulation of images, in *SIGIR 2021* (ACM, New York, 2021)
8. V.W. Anelli, Y. Deldjoo, T. Di Noia, A. Ferrara, F. Narducci, Federank: User controlled feedback with federated recommender systems, in *Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28—April 1, 2021, Proceedings, Part I*. Lecture Notes in Computer Science, vol. 12656 (Springer, Berlin, 2021), pp. 32–47
9. V.W. Anelli, Y. Deldjoo, T. Di Noia, A. Ferrara, F. Narducci, How to put users in control of their data in federated top-n recommendation with learning to rank, in ed. by C.-C. Hung, J. Hong, A. Bechini, E. Song, *SAC '21: The 36th ACM/SIGAPP Symposium on Applied Computing, Virtual Event, Republic of Korea, March 22–26, 2021* (ACM, New York, 2021), pp. 1359–1362
10. V.W. Anelli, Y. Deldjoo, T. Di Noia, F.A. Merra, Understanding the effects of adversarial personalized ranking optimization method on recommendation quality, in *AdvML 2021: 3rd Workshop on Adversarial Learning Methods for Machine Learning and Data Mining, Virtual Event, August 14–18, 202q* (2021)
11. M. Arjovsky, S. Chintala, L. Bottou, Wasserstein GAN. CoRR, abs/1701.07875 (2017)
12. L. Backstrom, J. Leskovec, Supervised random walks: Predicting and recommending links in social networks, in ed. by I. King, W. Nejdl, H. Li, *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9–12, 2011* (ACM, New York, 2011), pp. 635–644
13. G. Beigi, A. Mosallanezhad, R. Guo, H. Alvari, A. Nou, H. Liu, Privacy-aware recommendation with private-attribute protection using adversarial learning, in *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3–7, 2020* (2020), pp. 34–42
14. D. Berthelot, T. Schumm, L. Metz, BEGAN: boundary equilibrium generative adversarial networks. CoRR abs/1703.10717 (2017)
15. R. Bhaumik, C. Williams, B. Mobasher, R. Burke, Securing collaborative filtering against malicious attacks through anomaly detection, in *Proceedings of the 4th Workshop on Intelligent Techniques for Web Personalization (ITWP'06), Boston*, vol. 6 (2006), p. 10
16. B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Srndic, P. Laskov, G. Giacinto, F. Roli, Evasion attacks against machine learning at test time, in ed. by H. Blockeel, K. Kersting, S. Nijssen, F. Zelezný, *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23–27, 2013, Proceedings, Part III*. Lecture Notes in Computer Science, vol. 8190 (Springer, Berlin, 2013), pp. 387–402
17. B. Biggio, I. Corona, B. Nelson, B.I.P. Rubinstein, D. Maiorca, G. Fumera, G. Giacinto, F. Roli, Security evaluation of support vector machines in adversarial environments. CoRR abs/1401.7727 (2014)
18. J. Bourdeau, J. Hendler, R. Nkambou, I. Horrocks, B.Y. Zhao (eds.), *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11–15, 2016* (ACM, New York, 2016)
19. R. Burke, M.P. O'Mahony, N.J. Hurley, Robust collaborative recommendation, in ed. by Ricci et al., *Recommender Systems Handbook* (Springer, Berlin, 2015), pp. 961–995
20. Y. Cai, D. Zhu, Trustworthy and profit: a new value-based neighbor selection method in recommender systems under shilling attacks. Decision Support Syst. **124**, 113112 (2019)
21. J. Cao, Z. Wu, B. Mao, Y. Zhang, Shilling attack detection utilizing semi-supervised learning method for collaborative recommender system. World Wide Web **16**(5–6), 729–748 (2013)
22. S. Cao, N. Yang, Z. Liu, Online news recommender based on stacked auto-encoder, in ed. by G. Zhu, S. Yao, X. Cui, S. Xu, *16th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2017, Wuhan, China, May 24–26, 2017* (IEEE Computer Society, Washington DC, 2017), pp. 721–726

23. Y. Cao, X. Chen, L. Yao, X. Wang, W.E. Zhang, Adversarial attacks and detection on reinforcement learning-based interactive recommender systems, in J. Huang, Y. Chang, X. Cheng, J. Kamps, V. Murdock, J.-R. Wen, Y. Liu, *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25–30, 2020* (ACM, New Yrok, 2020), pp. 1669–1672
24. N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I.J. Goodfellow, A. Madry, A. Kurakin, On evaluating adversarial robustness. CoRR abs/1902.06705 (2019)
25. N. Carlini, D.A. Wagner, Defensive distillation is not robust to adversarial examples. CoRR abs/1607.04311 (2016)
26. N. Carlini, D.A. Wagner, Adversarial examples are not easily detected: Bypassing ten detection methods, in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017* (2017), pp. 3–14
27. N. Carlini, D.A. Wagner, Audio adversarial examples: Targeted attacks on speech-to-text, in *2018 IEEE Security and Privacy Workshops, SP Workshops 2018, San Francisco, CA, USA, May 24, 2018* (2018), pp. 1–7
28. A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, D. Mukhopadhyay, Adversarial attacks and defences: a survey. CoRR, abs/1810.00069 (2018)
29. P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, C.-J. Hsieh, ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models, in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017* (2017), pp. 15–26
30. H. Chen, J. Li, Adversarial tensor factorization for context-aware recommendation, in *RecSys* (ACM, New York, 2019), pp 363–367
31. H. Chen, J. Li, Data poisoning attacks on cross-domain recommendation, in ed. by W. Zhu, D. Tao, X. Cheng, P. Cui, E.A. Rundensteiner, D. Carmel, Q. He, J.X. Yu, *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3–7, 2019* (ACM, New York, 2019), pp. 2177–2180
32. L. Chen, Y. Xu, F. Xie, M. Huang, Z. Zheng, Data poisoning attacks on neighborhood-based recommender systems. CoRR abs/1912.04109 (2019)
33. P.-A. Chirita, W. Nejdl, C. Zamfir, Preventing shilling attacks in online recommender systems, in ed. by A. Bonifati, D. Lee, *Seventh ACM International Workshop on Web Information and Data Management (WIDM 2005), Bremen, Germany, November 4, 2005* (ACM, New York, 2005), pp. 67–74
34. K. Christakopoulou, A. Banerjee, Adversarial attacks on an oblivious recommender, in *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16–20, 2019*, (2019), pp. 322–330
35. C. Clavier, Secret external encodings do not prevent transient fault analysis, in ed. by P. Paillier, I. Verbauwhede, *Cryptographic Hardware and Embedded Systems:CHES 2007, 9th International Workshop, Vienna, Austria, September 10–13, 2007, Proceedings*. Lecture Notes in Computer Science, vol. 4727 (Springer, Berlin, 2007), pp. 181–194
36. H. Dai, H. Li, T. Tian, X. Huang, L. Wang, J. Zhu, L. Song, Adversarial attack on graph structured data, in ed. by J.G. Dy, A. Krause, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10–15, 2018*. *Proceedings of Machine Learning Research* PMLR, vol. 80 (2018), pp. 1123–1132
37. Y. Deldjoo, T. Di Noia, F.A. Merra, Adversarial machine learning in recommender systems (AML-RecSys), in *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3–7, 2020* (ACM, 2020), pp. 869–872
38. Y. Deldjoo, T. Di Noia, E.D. Sciascio, F.A. Merra, How dataset characteristics affect the robustness of collaborative recommendation models, in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25–30, 2020* (ACM, New York, 2020), pp. 951–960
39. Y. Deldjoo, M. Schedl, P. Cremonesi, G. Pasi, Recommender systems leveraging multimedia content. ACM Comput. Surv. **53**(5), 106:1–106:38 (2020)

40. Y. Deldjoo, V.W. Anelli, H. Zamani, A. Bellogín, T. Di Noia, A flexible framework for evaluating user and item fairness in recommender systems. User Model. User-Adapted Int. **31**, 457–511 (2021)
41. Y. Deldjoo, T. Di Noia, F.A. Merra, A survey on adversarial recommender systems: from attack/defense strategies to generative adversarial networks. *ACM Computing Surveys* **54**, 1–38 (2021)
42. Y. Du, M. Fang, J. Yi, C. Xu, J. Cheng, D. Tao, Enhancing the robustness of neural collaborative filtering systems under malicious attacks. IEEE Trans. Multimedia **21**(3), 555–565 (2019)
43. G.K. Dziugaite, Z. Ghahramani, D.M. Roy, A study of the effect of JPG compression on adversarial images. CoRR abs/1608.00853 (2016)
44. M. Fang, G. Yang, N.Z. Gong, J. Liu, Poisoning attacks to graph-based recommender systems, in *ACSAC* (ACM, 2018), pp. 381–392
45. M. Fang, N.Z. Gong, J. Liu, Influence function based data poisoning attacks to top-n recommender systems, in ed. by Y. Huang, I. King, T.-Y. Liu, M. van Steen, *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20–24, 2020* (ACM / IW3C2, New York/Geneva, 2020), pp. 3019–3025
46. C. Frederickson, M. Moore, G. Dawson, R. Polikar, Attack strength vs. detectability dilemma in adversarial machine learning, in *2018 International Joint Conference on Neural Networks, IJCNN 2018, Rio de Janeiro, Brazil, July 8–13, 2018* (IEEE, Piscataway, 2018), pp. 1–8
47. J. Gao, J. Lanchantin, M.L. Soffa, Y. Qi, Black-box generation of adversarial text sequences to evade deep learning classifiers, in *2018 IEEE Security and Privacy Workshops, SP Workshops 2018, San Francisco, CA, USA, May 24, 2018* (2018), pp. 50–56
48. M. Ge, C. Delgado-Battenfeld, D. Jannach, Beyond accuracy: Evaluating recommender systems by coverage and serendipity, in *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26–30, 2010* (2010), pp. 257–260
49. F. González, Y. Yu, A. Figueroa, C. López, C.R. Aragon, Global reactions to the cambridge analytica scandal: A cross-language social media study, in *WWW* (2019)
50. I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings* (2015)
51. S. Gu, L. Rigazio, Towards deep neural network architectures robust to adversarial examples, in *ICLR (Workshop)* (2015)
52. I. Gunes, C. Kaleli, A. Bilge, H. Polat, Shilling attacks against recommender systems: a comprehensive survey. Artif. Intell. Rev. **42**(4), 767–799 (2014)
53. R. He, J.J. McAuley, VBPR: Visual bayesian personalized ranking from implicit feedback, in ed. by D. Schuurmans, M.P. Wellman, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12–17, 2016, Phoenix, Arizona, USA* (AAAI Press, Palo Alto, 2016), pp. 144–150
54. X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in *WWW* (ACM, New York, 2017), pp. 173–182
55. X. He, Z. He, X. Du, T.-S. Chua, Adversarial personalized ranking for recommendation, in *SIGIR* (ACM, New York, 2018), pp. 355–364
56. S. Hidano, S. Kiyomoto, Recommender systems robust to data poisoning using trim learning, in ed. by S. Furnell, P. Mori, E.R. Weippl, O. Camp, *Proceedings of the 6th International Conference on Information Systems Security and Privacy, ICISSP 2020, Valletta, Malta, February 25–27, 2020*, SCITEPRESS (2020), pp. 721–724
57. R. Hu, Y. Guo, M. Pan, Y. Gong, Targeted poisoning attacks on social recommender systems, in *2019 IEEE Global Communications Conference, GLOBECOM 2019, Waikoloa, HI, USA, December 9–13, 2019* (IEEE, Piscataway, 2019), pp. 1–6
58. Y. Koren, R. Bell, Advances in collaborative filtering, in *Recommender Systems Handbook* (Springer, Berlin, 2015), pp. 77–118

59. A. Kurakin, I.J. Goodfellow, S. Bengio, Adversarial examples in the physical world, in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Workshop Track Proceedings* (2017)
60. A. Kurakin, I.J. Goodfellow, S. Bengio, Adversarial machine learning at scale, in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings* (2017)
61. S.K. Lam, J. Riedl, Shilling recommender systems for fun and profit, in *Proceedings of the 13th International Conference on World Wide Web, WWW 2004, New York, NY, USA, May 17–20, 2004* (2004), pp. 393–402
62. À. Lapedriza, H. Pirsiavash, Z. Bylinskii, A. Torralba, Are all training examples equally valuable? CoRR abs/1311.6510 (2013)
63. J. Lee, S. Abu-El-Haija, B. Varadarajan, A. Natsev, Collaborative deep metric learning for video understanding, in ed. by Y. Guo, F. Farooq, *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19–23, 2018* (ACM, New York, 2018), pp. 481–490
64. B. Li, Y. Wang, A. Singh, Y. Vorobeychik, Data poisoning attacks on factorization-based collaborative filtering, in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5–10, 2016, Barcelona, Spain* (2016), pp. 1885–1893
65. R. Li, X. Wu, W. Wang, Adversarial learning to compare: Self-attentive prospective customer recommendation in location based social networks, in *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3–7, 2020* (2020), pp. 349–357
66. Z. Liu, M.A. Larson, Adversarial item promotion: Vulnerabilities at the core of top-n recommenders that use images to address cold start. CoRR abs/2006.01888 (2020)
67. Y. Liu, X, Xia, L. Chen, X. He, C. Yang, Z. Zheng, Certifiable robustness to discrete adversarial perturbations for factorization machines, in ed. by J. Huang, Y. Chang, X. Cheng, J. Kamps, V. Murdock, J.-R. Wen, Y. Liu, *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25–30, 2020* (ACM, New York, 2020), pp. 419–428
68. A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings* (2018)
69. X. Mao, Q. Li, H. Xie, R.Y.K. Lau, Z. Wang, S.P. Smolley, Least squares generative adversarial networks, in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22–29, 2017* (2017), pp. 2813–2821
70. J.J. McAuley, C. Targett, Q. Shi, A. van den Hengel, Image-based recommendations on styles and substitutes, in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9–13, 2015* (2015), pp. 43–52
71. X. Meng, S. Wang, K. Shu, J. Li, B. Chen, H. Liu, Y. Zhang, Personalized privacy-preserving social recommendation, in *AAAI* (2018)
72. X. Meng, S. Wang, K. Shu, J. Li, B. Chen, H. Liu, Y. Zhang, Towards privacy preserving social recommendation under personalized privacy settings, World Wide Web **22**, 2853–2881 (2019)
73. B. Mobasher, R.D. Burke, R. Bhaumik, C. Williams, Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. ACM Trans. Int. Techn. **7**(4), 23 (2007)
74. P. Neekhara, S. Hussain, P. Pandey, S. Dubnov, J.J. McAuley, F. Koushanfar, Universal adversarial perturbations for speech recognition systems, in ed. by G. Kubin, Z. Kacic, *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15–19 September 2019*, ISCA (2019), pp. 481–485

75. T. Di Noia, D. Malitesta, F.A. Merra, Taamr: Targeted adversarial attack against multimedia recommender systems, in *50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops, DSN Workshops 2020, Valencia, Spain, June 29–July 2, 2020* (IEEE, 2020), pp. 1–8
76. M.P. O'Mahony, N.J. Hurley, G.C. M. Silvestre, Recommender systems: Attack types and strategies, in *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9–13, 2005, Pittsburgh, Pennsylvania, USA* (2005), pp. 334–339
77. N. Papernot, P.D. McDaniel, A. Swami, R.E. Harang, Crafting adversarial input sequences for recurrent neural networks, in ed. by J. Brand, M.C. Valenti, A. Akinpelu, B.T. Doshi, B.L. Gorsic, *2016 IEEE Military Communications Conference, MILCOM 2016, Baltimore, MD, USA, November 1–3, 2016* (IEEE, Piscataway, 2016), pp. 49–54
78. D.H. Park, Y. Chang, Adversarial sampling and training for semi-supervised information retrieval, in *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13–17, 2019* (2019), pp. 1443–1453
79. S. Rendle, L. Schmidt-Thieme, Pairwise interaction tensor factorization for personalized tag recommendation, in *WSDM* (ACM, New York, 2010), pp. 81–90
80. S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, in *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18–21, 2009* (2009), pp. 452–461
81. A.W. Rix, J.G. Beerends, M.P. Hollier, A.P. Hekstra, Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs, in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2001, 7–11 May, 2001, Salt Palace Convention Center, Salt Lake City, Utah, USA, Proceedings* (IEEE, Piscataway, 2001), pp. 749–752
82. A. Rozsa, E.M. Rudd, T.E. Boult, Adversarial diversity and hard positive generation, in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2016, Las Vegas, NV, USA, June 26–July 1, 2016* (IEEE Computer Society, Washington DC, 2016), pp. 410–417
83. L. Schönherr, K. Kohls, S. Zeiler, T. Holz, D. Kolossa, Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding, in *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24–27, 2019* (The Internet Society, Reston, 2019)
84. A.C. Serban, E. Poll, Adversarial examples: a complete characterisation of the phenomenon. CoRR abs/1810.01185 (2018)
85. A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J.P. Dickerson, C. Studer, L.S. Davis, G. Taylor, T. Goldstein, Adversarial training for free! in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8–14 December 2019, Vancouver, BC, Canada* (2019), pp. 3353–3364
86. Y. Shi, M. Larson, A. Hanjalic, Collaborative filtering beyond the user-item matrix: a survey of the state of the art and future challenges. ACM Comput. Surv. **47**(1), 3:1–3:45, (2014)
87. Y. Shi, Y. Sagduyu, A. Grushin, How to steal a machine learning classifier with deep learning, in *2017 IEEE International Symposium on Technologies for Homeland Security (HST)* (IEEE, Piscataway, 2017), pp. 1–5
88. J. Song, Z. Li, Z. Hu, Y. Wu, Z. Li, J. Li, J. Gao, Poisonrec: An adaptive data poisoning framework for attacking black-box recommender systems, in *36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, April 20–24, 2020* (IEEE, Psicataway, 2020), pp. 157–168
89. A.P. Sundar, F. Li, X. Zou, T. Gao, E.D. Russomanno, Understanding shilling attacks and their detection traits: a comprehensive survey. IEEE Access **8**, 171703–171715 (2020)
90. C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I.J. Goodfellow, R. Fergus, Intriguing properties of neural networks, in *ICLR* (2014)

91. J. Tang, X. Du, X. He, F. Yuan, Q. Tian, T. Chua, Adversarial training towards robust multimedia recommender system, in *IEEE Transactions on Knowledge and Data Engineering* **325**, 1–1 (2019)
92. J. Tang, H. Wen, K. Wang, Revisiting adversarially learned injection attacks against recommender systems, in *Fourteenth ACM Conference on Recommender Systems* (2020), pp. 318–327
93. J. Tang, X. Du, X. He, F. Yuan, Q. Tian, T.-S. Chua, Adversarial training towards robust multimedia recommender system. IEEE Trans. Knowl. Data Eng. **32**(5), 855–867 (2020)
94. N. Tintarev, J. Masthoff, Explaining recommendations: Design and evaluation, in ed. by Ricci et al., *Recommender Systems Handbook* (Springer, Berlin, 2015), pp. 353–382
95. T. Tran, R. Sweeney, K. Lee, Adversarial mahalanobis distance-based attentive song recommender for automatic playlist continuation, in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21–25, 2019* (2019), pp. 245–254
96. A. van den Oord, S. Dieleman, B. Schrauwen, Deep content-based music recommendation, in ed. by C.J.C. Burges, L. Bottou, Z. Ghahramani, K.Q. Weinberger, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5–8, 2013, Lake Tahoe, Nevada, United States* (2013), pp. 2643–2651
97. S. Vargas, P. Castells, Rank and relevance in novelty and diversity metrics for recommender systems, in ed. by B. Mobasher, R.D. Burke, D. Jannach, G. Adomavicius,*Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, October 23–27, 2011* (ACM, New York, 2011), pp. 109–116
98. Y. Vorobeychik, M. Kantarcioglu, *Adversarial Machine Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning (Morgan & Claypool Publishers, San Rafael, 2018)
99. J. Wang, P. Han, Adversarial training-based mean bayesian personalized ranking for recommender system. IEEE Access **8**, 7958–7968 (2020)
100. Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
101. T. Wang, J. Huan, B. Li, Data dropout: Optimizing training data for convolutional neural networks, in ed. by L.H. Tsoukalas, É. Grégoire, M. Alamaniotis, *IEEE 30th International Conference on Tools with Artificial Intelligence, ICTAI 2018, 5–7 November 2018, Volos, Greece* (IEEE, Piscataway, 2018), pp. 39–46
102. Z. Wei, J. Chen, X. Wei, L. Jiang, T.-S. Chua, F. Zhou, Y.-G. Jiang, Heuristic black-box adversarial attacks on video recognition models, in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7–12, 2020* (AAAI Press, Palo Alto, 2020), pp. 12338–12345
103. S. Winkler, P. Mohandas, The evolution of video quality measurement: From PSNR to hybrid metrics. IEEE Trans Broadcasting **54**(3), 660–668 (2008)
104. R.R. Wiyatno, A. Xu, O. Dia, A. de Berker, Adversarial examples in modern machine learning: a review. CoRR abs/1911.05268 (2019)
105. C. Wu, F. Wu, X. Wang, Y. Huang, X. Xie, Fairness-aware news recommendation with decomposed adversarial learning, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **35**(5), 4462–4469 (2021)
106. C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, A.L. Yuille, Adversarial examples for semantic segmentation and object detection, in *ICCV* (IEEE Computer Society, Washington, DC, 2017), pp. 1378–1387
107. Y. Xu, L. Chen, F. Xie, W. Hu, J. Zhu, C. Chen, Z. Zheng, Directional adversarial training for recommender systems, in *ECAI 2020* (2020)
108. G. Yang, N.Z. Gong, Y. Cai, Fake co-visitation injection attacks to recommender systems, in *NDSS* (2017)

109. F. Yuan, L. Yao, B. Benatallah, Adversarial collaborative auto-encoder for top-n recommendation, in *International Joint Conference on Neural Networks, IJCNN 2019 Budapest, Hungary, July 14–19, 2019* (2019), pp. 1–8
110. F. Yuan, L. Yao, B. Benatallah, Adversarial collaborative neural network for robust recommendation, in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21–25, 2019* (2019), pp. 1065–1068
111. X. Yuan, P. He, Q. Zhu, X. Li, Adversarial examples: attacks and defenses for deep learning. IEEE Trans. Neural Netw. Learning Syst. **30**(9), 2805–2824 (2019)
112. Q. Zhang, J. Wang, H. Huang, X. Huang, Y. Gong, Hashtag recommendation for multimodal microblog using co-attention network, in ed. by C. Sierra, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19–25, 2017*, ijcai.org (2017), pp. 3420–3426
113. L. Zheng, V. Noroozi, P.S. Yu, Joint deep modeling of users and items using reviews for recommendation, in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6–10, 2017* (2017), pp. 425–434
114. R. Zhang, P. Isola, A.A. Efros, E. Shechtman, O. Wang, The unreasonable effectiveness of deep features as a perceptual metric, in *CVPR 2018* (2018)
115. H. Zhang, Y. Li, B. Ding, J. Gao, Practical data poisoning attack against next-item recommendation, in *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20–24, 2020* (2020), pp. 2458–2464
116. W.E. Zhang, Q.Z. Sheng, A. Alhazmi, C. Li, Adversarial attacks on deep-learning models in natural language processing: a survey. ACM Trans. Intell. Syst. Technol. **11**(3), 1–41 (2020)
117. W. Zhou, J. Wen, Q. Xiong, M. Gao, J. Zeng, SVM-TIA a shilling attack detection method based on SVM and target item analysis in recommender systems. Neurocomputing **210**, 197–205 (2016)
118. W. Zhou, J. Wen, Q. Qu, J. Zeng, T. Cheng, Shilling attack detection for recommender systems based on credibility of group users and rating time series. PloS one **13**(5), e0196533 (2018)
119. Z. Zhu, J. Wang, J. Caverlee, Measuring and mitigating item under-recommendation bias in personalized ranking systems, in *SIGIR* (2020)