Adversarial Machine Learning in Recommender Systems

Adversarial Machine Learning in Recommender Systems

Felice Antonio Merra

Supervisor: Prof. Tommaso Di Noia

Coordinator: Prof. Mario Carpentieri

Department of Electric Engineering and Computer Engineering
Polytechnic University of Bari

This dissertation is submitted for the degree of
Doctor of Philosophy

Polytechnic University of Bari December 2021

Abstract

Recommender systems are ubiquitous. Our digital lives are influenced by their use when, for instance, we select the news to read, the product to buy, the friend to connect, and the movie to watch. While enormous academic research efforts have been mainly focused on getting high-quality recommendations to reach the maximum customers' satisfaction, little effort has been devoted to studying the integrity and security of these systems. Is there an underlying relationship between the characteristics of the historical user-item interactions and the efficacy of injection of false users/feedback strategies against collaborative models? Can public semantic data be used to perform attacks more potent in raising the recommendability of victim items? Can a malicious user (i.e., the adversary) poison or evade the image data of visual recommenders with adversarial perturbed product images? What is a possible defensive solution to reduce the effectiveness of test-time adversarial attacks? Is the family of model-based recommenders more vulnerable to multi-step gradient-based adversarial perturbations? Furthermore, is the adversarial training robustification still effective in the last scenario? Is this training defense influencing the beyond-accuracy and bias performance?

This dissertation intends to pave the way towards more robust recommender systems, beginning with understanding how a model can be made more robust, the cost of robustness in terms of recommendation quality, and the adversarial risks of modern recommenders. This thesis, getting inspiration from the literature on the security of collaborative models against the insertion of hand-engineered fake profiles and the recent advances of adversarial machine learning methods in other research areas like computer vision, contributes to several directions: (i) the proposal of a practical framework to interpret the impact of data characteristics on the robustness of collaborative recommenders, (ii) the design of powerful attack strategies using publicly available semantic data, (iii) the identification of severe adversarial vulnerabilities of visual-based recommender models where adversaries can break the recommendation integrity by pushing products to the highest recommendation positions with a simple and human-imperceptible perturbation of products' images, (iv) the design of a novel defense method to protect visual recommenders against test-time adversarial attacks,

(v) the proposal of robust adversarial perturbation methods capable of completely breaking the accuracy of matrix factorization recommenders, and (vi) a formal study that examines the effects of adversarial training in reducing the recommendation quality of state-of-the-art model-based recommenders.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Recommendation systems (RSs) have become a necessary component of our everyday digital lives, freeing our minds from the superabundance of products and services attainable on online platforms. Amazon [150], Google [68], Spotify [108], and Netflix [40] are de facto standard examples of how much gig companies take advantage of RSs to make as much personalized as possible customers' experiences.

The keystone of good recommendations lies in the use of machine learning (ML) techniques to extrapolate behavioral patterns from historical users' interactions, e.g., purchased products, watched movies, and visited restaurants, and assist users' decision-making processes by curating a list of items that the user would be interested in. Additionally, highly qualitative personalization has also been reached by exploiting the taste similarities between the users in the platform. This approach, known as collaborative filtering (CF), dominates the scene from the origins of the research on recommender systems [191].

When learning to recommend, the first assumption is that all the platform entities, e.g., customers, sellers, and content editors, are honest and have trustful behaviors. This is far from the truth. There are many facets of the security of the recommendation process which are pretty under-investigated. Thus, considering the terrific benefits of RSs on increasing sales and supporting users, there is a largely untapped territory for investigating the safety of RSs against adversaries having an incentive to compromise the functionalities of ML-based RSs, which this dissertation endeavors to shed light on.

Adversarial machine learning (AML) is the research field investigating the vulnerabilities inherent to ML systems' design and the means to defend against them. A noticeable hype on the security of ML models hiked up after the presentation of worrying real-world examples on the fact that traffic-sign ML-based classifiers, used in autonomous vehicles, would be easily fooled by human-imperceptible (adversarial) perturbations of

traffic signals [101, 160]. From 2017 adversarial techniques have gained attention in recommendation scenarios. We provide the literature review about the application of AML in RSs in Chapter 2.

RSs face two comprehensive examples of risks: integrity and availability. Breaking the integrity means the adversary induces a model output (i.e., the recommendations) different from the original one. For example, adversarial attacks attempt to push/target (victim) items into high/low positions in the recommendation lists. Chapters 3 to 6 present our research contributions to this issue. Then, compromising the availability involves scenarios that the malicious user attempts to reduce the recommendation quality (e.g., the accuracy of top-K recommendations). For instance, based on the level of knowledge of the victim recommender, the attacker can try to destroy the accuracy of the model, making the recommendation lists completely unuseful with a consequent reduction of the users' trust towards the platform. In Chapter 7, we present novel algorithms for crafting adversarial examples to destroy the availability of standard recommendation techniques, and in Chapter 8, we put on a formal analysis on the influence on the RS availability of state-of-the-art adversarial protection of recommender systems.

1.1 Thesis Statement

This work characterizes and undertakes adversarial risks in the recommender system research domain to assess and improve our understanding of deployed ML-based RSs security. As mentioned earlier, our research interest has focused on the integrity and the availability of recommendation models in adversarial settings.

Starting from the foundations of the recommender system and adversarial machine learning, we provide an in-depth literature review on the existing works and the preliminaries proper to place our research contributions summarized later in Section 1.2. We organize the adversarial techniques to reach the malicious goals with the following strategies: injection of hand-engineered and machine-learned fake profile (known as Shilling Profiles), noise added to the recommender's machine-learned parameters, and human-imperceptible perturbation of content data used in content-based and hybrid recommenders.

Paying attention to the first strategy, we investigate whether the recorded set of user-to-item recorded interaction characteristics can influence the efficacy of the injected fake profiles. The intuition is to propose an easy-interpretable model that supports system designers in robustifying the model from a dataset perspective. This

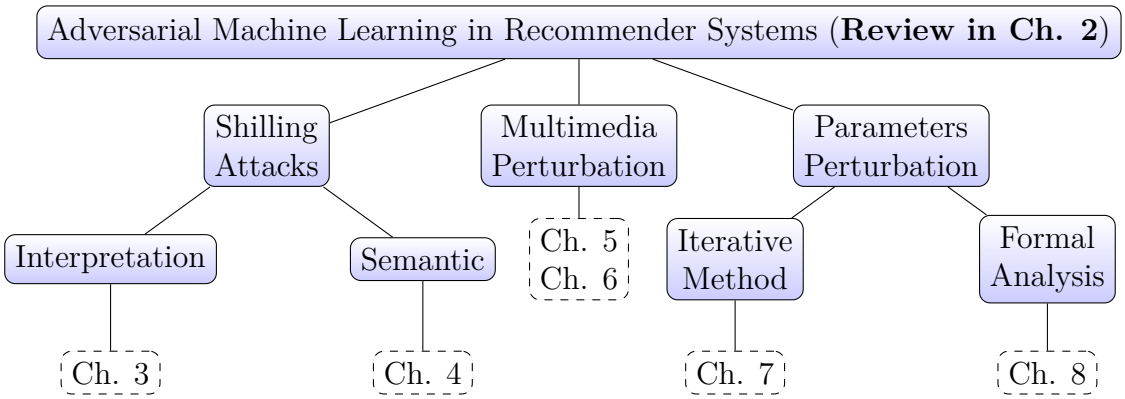


Fig. 1.1 Thesis organization.

part of our research has also been pursued by proposing a novel attack approach exploiting publicly available semantic information (e.g., knowledge graphs) to empower adversarial with limited or absent knowledge.

The second core of the investigation is related to the security of multimedia recommender models. Here, we investigate a new research problem related to studying the effects of adversarial examples crafted on item images on the reliability of a visual-based recommender model. A vast set of standard and novel adversarial attack and defense strategies in training and testing time settings have been analyzed and proposed to break and/or protect a large set of state-of-the-art visual recommenders.

Regarding the study of adversarial noise added to the model parameters, our research efforts have been focused on two main arguments. The former proposes a novel noising technique stronger than the existing ones affecting the model's availability (i.e., the recommender starts to behave randomly). The latter opens the investigations of the effects of state-of-the-art defense strategies on performance quality over accuracy.

In what follows, we detail the research contributions.

1.2 Research Contributions

The thesis discusses the research questions regarding how recommendation systems can be victims of adversaries and could be protected through the perspectives of novel adversarial machine learning techniques. Figure 1.1 presents an overview of our research arguments with the link to the chapters of this dissertation. Each part views the adversarial learning applications on different attack types (and defense) against a recommender model. The following sections provide additional details on this thesis's

research goals and contributions based on this structure. Note that, **Merra Felice Antonio** is the **corresponding author** of the scientific publications related to the research contributions presented in this dissertation [1].

1.2.1 Ch. 2: Survey, tutorials, and a book chapter on AML in RSs.

Unlike the following subsections, here, we do not present overall research questions related to the particular research contribution since the current section describes the surveying contribution of the application of AML in recommendation settings.

Contributions. While there exist several survey articles on general RS topics, for example Ekstrand et al. [90], Shi et al. [200], Quadrana et al. [183], we found a lack of literature reviews focusing on the application of AML techniques in the recommendation task. Motivated by this absence, we provide a comprehensive literature review by identifying, first, that the applications of AML have to be specifically referred to security aspects and not a novel recommendation algorithm based on generative adversarial network (GAN). Then, we propose an attack/defense-driven classification of the state-of-the-art adversarial applications whose has been at the core of our investigation and have motivated the research contributions described below.

Publications. The content of the foundations and review presented in Chapter 2 has been presented in the journal paper "*A Survey on Adversarial Recommender Systems: From Attack/Defense Strategies to Generative Adversarial Networks*" [82] published by ACM Computing Surveys (CSUR) and the book chapter "*Adversarial Recommender Systems: Attack, Defense, and Advances*" [20] accepted for publication in the 3rd Edition of Recommender Systems Handbook. Additionally, we have presented the content of these publications in three conference tutorials at WSDM2020 [81], RecSys2020 [19], and ECIR2021 [122], and during the summer internship held in Amazon.com.

Role of Ph.D. Candidate. Corresponding author of previous contributions, i.e., survey [82], tutorials [81, 19, 122], and book chapter [20].

[1] The authors of the publications are alphabetically ordered. The corresponding authors of publications are reported in the articles.

1.2.2 Ch. 3: Interpretation of the Impact of Data Characteristics on Robustness.

Is there an underlying relationship between the dataset characteristics computed on the matrix of recorded user-item interactions and the effectiveness of shilling attack against collaborative recommender models?

Contributions. In Chapter 3 of this dissertation, we present a systematic, in-depth exploratory research and analysis of the impact of dataset characteristics on the robustness performance of popular CF models subjected to famous shilling attack strategies. Mainly, we propose a *regression-based explanatory* framework to investigate the correlation between a suite of structural and value-based data characteristics extracted from the user-item feedback matrix (UIFM) and the robustness of CF models. Results of extensive experiments provide sufficient statistical evidence to accept the hypothesis that, first, the identified data characteristics can account for a considerable portion of variations in attack performance (global perspective) and, second, that there remain considerable differences in the significance (and directionality) of this impact among the characteristics.
Publications. The preliminary contributions to the effects of dataset characteristics on attacks efficacy appeared as the publication "*Assessing the Impact of a User-Item Collaborative Attack on Class of Users*" in the Workshop on the Impact of Recommender Systems held in conjunction with the 13th ACM Recommender Systems Conference (RecSys) 2019 [80]. Starting from the research contributions and open challenges of this article, we presented the regression framework in the publication "*How Dataset Characteristics Affect the Robustness of Collaborative Recommendation Models*" presented as a long paper at the 43rd ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR) 2020 [76]. A condensed version of the work has been presented at the 11th edition of the Italian Information Retrieval Workshop (IIR) 2021 in the published discussion paper titled "*A Regression Framework to Interpret the Robustness of Recommender Systems Against Shilling Attacks*" [77].
Role of Ph.D. Candidate. Corresponding author of all the published articles [80, 76, 77].

1.2.3 Ch. 4: Semantics-aware Shilling Attacks.

Can public available semantic information be exploited to develop more effective shilling attacks against CF models, where the effectiveness is measured in terms of a raise of the recommendability of the target items in the recommendation lists?

Contributions. In Chapter 4, we present a set of methods leveraging semantic-encoded information extracted from publicly available information resources obtained from KGs to generate more influential fake profiles that can undermine the performance of CF models. In this line of research, we focus on empowering adversaries' capabilities in breaking the integrity of a CF-RS without providing any additional information about the system. We propose a new technique, semantics-aware shilling attack *SAShA*, completely integrable with existing shilling attack strategies. Experiments in real-world datasets show that integrating *SAShA* with standard shilling attack strategies confirms that it is a powerful tool to implement effective attacks also when attackers do not have any knowledge of the victim RS. Additionally, we investigate the method efficacy changing the type of semantic information, the extraction depth on public knowledge graphs, and the algorithms used to evaluate the semantic similarities of target victims with the other items in the catalog.

Publications. The research contributions presented in this chapter are based on the conference articles "*SAShA: Semantics-Aware Shilling Attacks on Recommender Systems Exploiting Knowledge Graphs*" [16] and "*Knowledge-enhanced Shilling Attacks for Recommendation*" [21] presented at The Semantic Web - 17th International Conference (ESWC) 2020 and the 28th Italian Symposium on Advanced Database Systems (SEBD) 2020, respectively. Additionally, we have been invited to extend the method to the Semantic Web Journal. The article named "*Semantics-Aware Shilling Attacks against collaborative recommender systems via Knowledge Graphs*" is currently *under review* and publicly accessible on the journal platform [2].

Role of Ph.D. Candidate. Corresponding author of all the presented publications [16, 21].

1.2.4 Ch. 5: Poisoning of Multimedia Recommender Systems with Adversarial Images: Attacks and Defenses.

Can an adversary poison the data of multimedia recommender systems with adversarial samples? Do adversarial perturbations of product images confuse multimedia recommenders? Can we protect the model integrity?

Contributions. Most recommendation systems use multimedia content associated with products, e.g., images, videos, and descriptions, to empower the recommendation quality of collaborative recommender systems [83]. Among them, visual-based recommender systems (VRSs) have merged as powerful techniques thanks to the representational

[2] http://www.semantic-web-journal.net/system/files/swj2735.pdf

power of deep neural networks (DNNs) in capturing characteristics and semantics of the images to be integrated within the training of the recommendation model. As mentioned earlier, suppliers of the items on a recommendation platform can have malicious objectives and, in this line of research, we investigate adversarial settings where sellers can upload adversarial perturbed images of their items to damage the integrity of the model and push (or nuke) their frequency of recommendation in high positions. For instance, a malicious seller might upload images of socks products maliciously perturbed to be treated by an ML model as t-shirts (a popular bought product) in product recommendation. This action might push up the socks' product in high recommendation positions. Motivated by this case, we propose a set of attack strategies that have been demonstrated to break the model's efficacy through extensive experiments on real-world product recommendation datasets and several VRSs. Then, we investigate and experiment with defense solutions, showing a partial efficacy and the need for further exploration of this completely new adversarial scenario. Starting from these results, we have started to investigate the possibility of introducing a denoiser module that, independently of the visual recommender, can remove the adversarial noise from the input samples.

Publications. The first articles that put the foundation of the research direction taken in Chapter 5 are "*TAaMR: Targeted Adversarial Attack against Multimedia Recommender Systems*" [85] published at Dependable and Secure Machine Learning Workshop Co-located with the 50th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN) 2020 and "*Assessing Perceptual and Recommendation Mutation of Adversarially-Poisoned Visual Recommenders*" [27] presented at the 1st Workshop on Dataset Curation and Security co-located with the 34th Conference on Neural Information Processing Systems (NeurIPS) 2020. Starting from the preliminary results obtained din the previous two articles, the framework and experimental results presented in the chapter appeared as a long paper in the proceedings of the 44th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR) 2021 [17].

Role of Ph.D. Candidate. Corresponding author of all the above-mentioned articles [85, 27, 17].

1.2.5 Ch. 6: Evading Multimedia Recommender Systems with Adversarial Images: Attacks and Defenses.

Can Adversarial Image Denoiser (AiD) reduce the effectiveness of adversaries that use test-time adversarially-perturbed product images? How much AiD application is affecting the overall accuracy and beyond-accuracy performance?

Contributions. Test-time (evasive) adversarial attack strategies have recently unveiled severe security issues against visually-aware recommender models. Indeed, adversaries can harm the integrity of recommenders by uploading item images with human-imperceptible adversarial perturbations capable of pushing a target item into higher recommendation positions. Under this class of attacks, we have focused our research interest on two main contributions: identifying the popularity influence on the attack efficacy and proposing the first test-time defensive method. As for the first research contribution, given the importance of items' popularity on the recommendation performance, in our research interest, we evaluate whether items' popularity influences the attacks' effectiveness. To this end, we have performed three state-of-the-art adversarial attacks against VBPR (a standard VRS) by varying the adversary knowledge (white- vs. black- box) and capability (the magnitude of the perturbation). The results obtained evaluating attacks on two real-world datasets have shed light on the remarkable efficacy of the attacks against the least popular items' opening novel open challenges on the importance of considering the popularity also in defensive settings. Regarding the second main contribution, to which we dedicate more attention in Chapter 6, we propose "Adversarial Image Denoiser" (AiD), a novel defense method to protect VRSs against adversarial attacks. In AiD, we exploit the idea of cleaning up the product images by the perturbations added by the adversaries. In particular, we propose a U-Net-based denoising autoencoder trained to minimize the visual differences between clean and adversarial images while preserving the recommender systems' behavior in clean settings. To verify the efficacy of the proposed defense solution, we have investigated the defense performance on three real-world datasets and two popular visual recommender models, one of which implements the state-of-the-art defensive solution (i.e., adversarial training) under three attack strategies (i.e., one black-box and two white-box). The experiments confirm that AiD is an effective solution for protecting visual recommender models against the set of tested attacks, reducing their effectiveness in varying the predicted preference scores and the target items' positions in the recommendation lists.

Publications. Being the effectiveness of test-time (evasion) attacks have already been analyzed in the original attack proposal articles [85, 67, 154], our initial research attention has been devoted to investigating whether items' popularity bias would have affected the efficacy of adversarial attacks on visual-based recommenders. Preliminary results on this novel line of research have been presented in the article "*Adversarial Attacks against Visual Recommendation: an Investigation on the Influence of Items' Popularity*" published at the 2nd Workshop on Online Misinformation- and Harm-Aware Recommender Systems in conjunction with the 15th ACM Conference on Recommender Systems [23]. The research article related to the proposal of a novel defense strategy to protect a visual-based recommender against adversarial perturbed images is currently under review as a long paper with the title "*Adversarial Image Denoiser to Defend Visual-based Recommender Systems against Attacks*". This last article is at the core of Chapter 6.

Role of Ph.D. Candidate. Corresponding author of the article referenced at [23] and the research contribution presented in a paper under review whose content is presented in Chapter 6.

1.2.6 Ch. 7: Iterative Methods to Perturb the Parameters of an RS.

Considering the parameters' instability to adversarial perturbation on model-based RSs, how vulnerable are the parameters to iterative gradient-based adversarial methods? Is the adversarial training approach working in robustifying the model against this attack?

Contributions. Inspired by recent studies showing that model-based recommender systems are not robust to adversarial perturbation of model parameters [115], which consists of the addition of minimal noise to the RS embeddings to crack the model availability, we propose gradient-based iterative methods. The research goal is to understand if the performance worsening caused by state-of-the-art perturbations can even be empowered with multi-step optimization techniques. Experiments show that the proposed strategies are the most powerful ones under a fixed perturbation budget (the maximum variation of model parameters caused by the addition of noise). Then, we verify that the proposal degrades accuracy and beyond-accuracy recommendation quality so much to make the victim model worse than a random (not-personalized) model, staying still effective also against adversarially trained models.

Publications. Chapter 6 is extracted from the article "MSAP: Multi-Step Adversarial Perturbations on Recommender Systems Embeddings" [12] published in the proceedings of the 34th International FLAIRS Conference Proceedings (FLAIRS) 2021.
Role of Ph.D. Candidate. Corresponding author of the published contribution [12].

1.2.7 Ch. 8: A Formal Analysis of Recommendation Quality of Adversarially Trained Recommenders

Since adversarial training has been demonstrated to disturb the model accuracy in the image classification task, how does it influence the recommendation performance on accuracy and beyond-accuracy perspectives?

Contributions. Adversarial personalized ranking (APR) is an adversarial training procedure proposed in [115] to robustify Bayesian personalized ranking [188], the most popular learning-to-rank optimization framework, against the injection of adversarial noise (the core attack in Chapter 7). Considering the performance alteration of adversarially trained classifiers for the image classification task [184], we focus on investigating the learning differences between APR and BPR to understand if APR could affect the recommendation quality. The proposed formal analysis shows that APR could be affected by amplifying popularity bias and reducing beyond-accuracy measures. The experimental results on five recommendation datasets on matrix factorization (MF) recommenders confirm this worsening of recommendation quality, motivating the design of novel robust learning procedures that can strike a more meaningful balance between accuracy, beyond accuracy, and low amplification of popularity bias.
Publications. This complete version of this work is currently under review. An initial contribution has been presented at the 3rd Workshop on Adversarial Learning Methods for Machine Learning and Data Mining in conjunction with the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) 2021 in the article "*Understanding the Effects of Adversarial Personalized Ranking Optimization Method on Recommendation Quality*" where has been awarded by the "MIT-IBM Watson AI Lab Best Paper Award." Then, the indexed articles are *"The Idiosyncratic Effects of Adversarial Training on Bias in Personalized Recommendation Learning"* published at RecSys 2021 [29] and *"A Formal Analysis of Recommendation Quality of Adversarially-trained Recommenders"* published at CIKM 2021 [28]. The last article has been nominated as "Runner-Up Best Short Paper".
Role of Ph.D. Candidate. Corresponding author of the accepted contributions [29, 28].

1.3 Bibliographical Notes

This section describes the research articles published during the Ph.D. but not profoundly discussed in the dissertation. Indeed, the following works have been conducted as simultaneous topics whose research questions have been raised while studying the literature.

For the theme of multimedia recommender systems, motivated by the effects of adversarial attack and defense performed against the DNN used in the system, we explored the effects of varying the DNN used to extract the visual features. The article "*A Study on the Relative Importance of Convolutional Neural Networks in Visually-Aware Recommender Systems*" [78] appeared in the 4th CVPR Workshop on Computer Vision for Fashion, Art, and Design, proved that a deeper feature extraction model, i.e., ResNet50 [112], ensures high accuracy and beyond-accuracy recommendation performance. An additional work published on visual recommender systems is "*Leveraging Content-Style Item Representation for Visual Recommendation*" accepted at the 44th European Conference on Information Retrieval [79], in which a novel visual attention mechanism has been proposed to enhance the performance in the visual-based recommendation task.

Regarding the research topic of adversarial machine learning, we investigated gradient-based perturbations on model parameters in MF-based link prediction methods. The research paper titled "*AMFLP: Adversarial Matrix Factorization-based Link Predictor in Social Graphs*" [73] published in the proceedings of the 29th Italian Symposium on Advanced Database Systems (SEBD) 2021, proposes a perturbation technique able to reduce the link prediction performance drastically and an adversarial training solution reducing this deterioration.

Additionally, we have co-authored the reproducibility framework presented in the resource paper "*Elliot: a Comprehensive and Rigorous Framework for Reproducible Recommender Systems Evaluation*" [13] published at SIGIR 2021. The framework makes more than 50 recommendation models available, including adversarial and GAN-based implementations, together with a large set of evaluation metrics, hyper-parameters strategies, and data-pre-processing operations to support easy-to-run and reproducible experiments for both researchers and industrial practitioners. The system is publicly available in a GitHub repository [3]. A demonstration paper fully dedicated to the integration of visual-based recommenders has been presented at RecSys 2021 in the indexed article named "*V-Elliot: Design, Evaluate and Tune Visual*

[3]https://github.com/sisinflab/elliot/

Recommender Systems" [15], while an extended abstracted, named "*How to perform reproducible experiments in the ELLIOT recommendation framework: data processing, model selection, and performance evaluation*", has been presented at the 11th edition of the Italian Information Retrieval Workshop (IIR) 2021 [14].

Lastly, I have co-authored with Jacek Golebiowski and Felix Biessmann the work titled "*Search Filter Recommendation using Language-Aware Label Embeddings*" an applied research paper under review that presents the research contributions reached during my Ph.D. internship at Amazon.com. The contributions of this work are related to the proposal of a novel deep learning model to recommend the most relevant set of product categories for each typed query.

Next, we present the background, preliminaries, and literature review of AML applications in RSs. Then, moving from Chapter 3 to Chapter 8, we detail the research contributions shown in Figure 1.1. Finally, we review the findings in this dissertation and propose several open problems and potential future work.

Chapter 2

Foundations and Background

We present a brief overview of the background concepts used throughout this thesis. In particular, we start from the foundations of recommender systems and adversarial machine learning presenting before citing and classifying the different kinds of adversarial learning applications in recommendation scenarios. Note that We will go in-depth in the related chapters for each field where We focused our research contributions.

The current chapter presents the terminology used throughout the remainder of this thesis. In general, this dissertation follows the convention: capital calligraphic (e.g., $\mathcal{A}$) to denote a set, bold uppercase (e.g., $\mathbf{X}$) to indicate a matrix, lowercase bold (e.g., $\mathbf{x}$) to express a vector, and simple lowercase (e.g., x) to denote a scalar.

2.1 Foundations of RS

Recommender systems have terrifically taken over online shopping by providing users with personalized recommendations to disentangle the chaotic flood of products on e-commerce platforms. They model consumers' preferences by learning from past behavioral data like rated, bought, or reviewed products. Collaborative filtering recommendation models play a pivotal role in online services in increasing traffic and promoting sales. They are widely adopted by various e-commerce and consumer-oriented services to recommend a whole range of items, including products, music, movies, news articles, friends, restaurants, and various others. Their basic assumption is that users who shared similar preferences in the past will likely agree in the future as well. Then, from an algorithmic point of view, these models keep track of users' interactions to find similarities in users' behavioral patterns. This dissertation will be focused only on collaborative recommender models.

I indicate with as $\mathcal{U}$ the set of users in the system, where $|\mathcal{U}|$ is the number of users. We denote with $\mathcal{I}$ the items set whose size is defined as $|\mathcal{U}|$. The preference score of a user $u \in \mathcal{U}$ on an item $i \in \mathcal{I}$ is a scalar denoted as $s_{ui} \in \mathbf{S}$, where $\mathbf{S} \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{I}|}$ is the sparse matrix of all possible user-item preference scores. The user-item preference score can be an explicit feedback (e.g., $s_{ui} \in \{1,2,3,4,5\}$ depending from the number of stars left by u on the bought product i), or an implicit feedback (e.g., $s_{ui} = 1$ is u bought i). We denote with $\mathcal{R}$ the set of (u,i) pairs for which s_{ui} is known and therefore $|\mathcal{R}|$ represents the total number of feedback recorded on a platform (i.e., the size of the recommendation dataset).

The recommendation problem can be defined as finding a utility function to automatically predict how much a user will like an item that is unknown to her (an unknown user-item preference score).

Definition 1 (Recommendation Problem)**.** *Given a utility function, $\hat{s} : \mathcal{U} \times \mathcal{I} \to \mathbb{R}$, the **Recommendation Problem** is defined as*

$$\forall u \in \mathcal{U}, i' = \underset{i \in \mathcal{I}}{\operatorname{argmax}}\, \hat{s}(i \mid u) \tag{2.1}$$

with $i' \in \mathcal{I}/\mathcal{I}_u^+$ is not in the list of (positive) items already seen by the user u (i.e., $\mathcal{I}_u^+$).

The solution to a recommendation problem heavily depends on the selected utility function $\hat{s}$ —usually, but not necessarily, a machine learning model— and on the information encoded within the dataset represented by $\mathcal{R}$.

Additionally, a common approach to address the Recommendation Problem is to present a personalized list of relevant items to each user in the platform. This problem can be modeled as a Ranking Task, and it is defined below.

Definition 2 (Ranking Task)**.** *Given a user $u \in \mathcal{U}$, the rank of a not-interacted item $i \in \mathcal{I}$ is defined via the bijective function in $\mathcal{I}$ as $\hat{s}(i|u)$. Let $\hat{r}(\cdot)$ be the ranking function based on the predicted value of the preference score function $\hat{s}(\cdot|\Theta)$, where Θ represents the ML recommender's model parameters. The **Ranking Task** builds a top-K recommendation list associated with the user u as follows,*

$$\hat{r}(i \mid u) := \left\{ |\{j : \hat{s}(j \mid u) \geq \hat{s}(i \mid u)\}|, i, j \in I \backslash I_u^+ \right\} \tag{2.2}$$

where I_u^+ is the list of (positive) items already seen by the user u.

The open nature of the collection of feedback makes the recommendation problem vulnerable to the injection of malicious users [105]. This dissertation explores in

Chapters 3 and 4 two adversarial settings related to the addition of fake profiles, named shilling attacks. Section 2.3.2 presents the background knowledge of the beforementioned attack strategy.

2.1.1 Recommendation Methods

The research interest in generating personalized lists of relevant items is the core challenge in the recommendation domain [139]. Recommendation techniques are generally classified into collaborative filtering (CF) [72, 188], content-based filtering (CBF) [181], and hybrid [190, 4].

CF leverages users' collective behavior data such as interactions and stated preferences to compute recommendations. CBF models recommend items similar to those preferred in the past based on the item's characteristics (e.g., item content). Finally, hybrid models combine CF and CBF techniques under a unique framework. In this thesis, we focus on collaborative filtering principles, which exploit the wisdom of crowds to empower modern recommenders, and hybrid recommenders, which exploit users' or items' additional data to get profits to form both collaborative and content-based signals. To set the background of recommendation techniques to investigate in the following chapters, we present the main approaches of CF models.

Collaborative Filtering (CF)

CF-RSs can be further categorized in two classes of models: neighborhood-based [98, 118] and model-based [188, 218]. **Neighborhood-based recommenders**, also known as memory-based recommenders, rely on computing similarities from user behavioral data (i.e., user-user or item-item similarities) to predict unknown user preferences. **Model-based recommenders** transform items and users into a shared latent factor space whose interactions explain the observed interactions. Depending on the type of interaction, model-based CF can be for example classified according to linear approaches, e.g., matrix factorization (MF) [188], and non-linear models, e.g., neural matrix factorization (NeuMF) [116]. Considering the popularity of MF-based solutions to implement recommendation systems, we have investigated ML-based solutions in any research contribution that will be detailed in Chapters 3 to 8. Below, we formally present the simplest MF model.

Matrix Factorization (MF). MF is a latent factor model that learns the *linearity* of the unknown preferences. It represents both items and users by vectors of latent

factors. These factors are learned from linear patterns of the user-item feedback matrix. The learned user and item embeddings are two low-rank matrices, one for the users, i.e., $\mathbf{P} \in \mathbb{R}^{|\mathcal{U}| \times h}$, and another for the items, i.e., $\mathbf{Q} \in \mathbb{R}^{|\mathcal{I}| \times h}$, where $h \ll |\mathcal{I}|, |U|$ and $h \in \mathbb{R}$ is the dimension of the embeddings. In MF, the model parameters Θ are $\{\mathbf{P}, \mathbf{Q}\}$. The preference prediction function is $\hat{s}(i|u) := \mu + b_u + b_i + \mathbf{q}_i^T \mathbf{p}_u$, where μ, b_u, and b_i are the overall average score, the observed bias of user u and item i, respectively, and $\mathbf{q}_i^T \mathbf{p}_u$ is the dot-product between the user, i.e., $\mathbf{p}_u \in \mathbf{P}$, and the item, i.e., $\mathbf{q}_i \in \mathbf{Q}$, embeddings.

2.1.2 Evaluation

As shown in Definition 2, the recommendation problem is solved presenting to each user u a recommendation list by sorting all the unrated items (i.e., $\mathcal{I}_u^- := \{i' \in \mathcal{I} / \mathcal{I}_u^+)$ by decreasing values of inferred preference score $\hat{s}(\cdot)$. We evaluate this list checking whether a part of ground truth interactions s_{ui} placed in the test set built on $\mathcal{R}$ have been covered in the first K positions, where $K \in \mathbb{N}$ is the threshold at which we evaluate the ranked list of products. From now on, we use top-K to indicate the first K recommended items. In what follows, we report the evaluation metrics capturing the performance of an RS that we will use in chapters of this dissertation. If a novel metric has been proposed in a publication, it will be presented in the related chapter.

Accuracy Metrics

Below, we define the main accuracy metrics used to evaluate the performance of a recommender model. Note that the following measures are all defined in the $[0, 1]$-range, where is the best possible metric value.

Definition 3 (Precision (Pr@K)). *Let Rel_u bet the set of items relevant to user $u \in \mathcal{U}$, and Rec_u is the top-K list of items recommended to u.*

$$\text{Pr@K} = \frac{1}{|\mathcal{U}|} \sum_{u=1}^{\mathcal{U}} \frac{|Rel_u \cap Rec_u|}{|Rec_u|} \tag{2.3}$$

Pr@K *is the fraction of previously interacted items correctly inserted in the topK recommendation list.*

Definition 4 (Recall (Re@K)).

$$\text{Re@K} = \frac{1}{|\mathcal{U}|} \sum_{u=1}^{\mathcal{U}} \frac{|Rel_u \cap Rec_u|}{|Rel_u|} \tag{2.4}$$

Re@K *is the fraction of the relevant items that are successfully retrieved.*

Definition 5 (Hit Ratio (HR@K))**.** *Let $hit(u, K)$ a binary function that is one if at list one recommended item has been interacted by u, 0 otherwise, then the Hit Ratio at K is defines as follows,*

$$\text{HR@K} = \frac{1}{|\mathcal{U}|} \sum_{u=1}^{\mathcal{U}} hit(u, K) \tag{2.5}$$

HR@K *compares the top-K recommendations for each user u to the recorded ones (e.g., the interaction stored in the test set). If they match, then increase the hit rate by 1.*

Definition 6 (normalized Discounted Cumulative Gain (nDCG@K))**.** *Let rel_{u,i_k} the gain that u would get when the item i is recommended in the position $k \in K$ of the recommendation list. Let $2^{rel_{u,i_k}}$ be equals to 1 if the item hits, otherwise 0. Then, following [104], the nDCG@K is defined as follows,*

$$\begin{aligned}
\text{DCG}_\text{u}\text{@K} &= \sum_{k=1}^{K} \frac{2^{rel_{u,i_k}} - 1}{\log_2(k+1)} \\
\text{IDCG}_\text{u}\text{@K} &= \sum_{k=1}^{K} \frac{1}{\log_2(k+1)} \\
\text{nDCG}_\text{u}\text{@K} &= \frac{DCG_u@K}{IDCG_u@K} \\
\text{nDCG@K} &= \frac{1}{|\mathcal{U}|} \sum_{u=1}^{\mathcal{U}} \text{nDCG}_\text{u}\text{@K}
\end{aligned} \tag{2.6}$$

where IDCG_u@K *is the ideal $DCG_u@K$ which represent the ideal order that the recommended items should follow.*

Beyond-Accuracy Metrics

Due to the large impact of RSs in the society [35, 36], a huge research effort has been dedicated to beyond-accuracy objectives [44]. For instance, studying whether the suggested items are novel and cover the complete catalog, and proposing methods to mitigate several types of biases [60], e.g., selection bias[193], exposure bias [164], and popularity bias [1, 3]. To measure the beyond-accuracy performance, we most used metric in this dissertation if the item coverage ($Cov_{\%}@k$). Beyond-accuracy metrics used in the experimental section and not described in this chapter will be presented there.

Definition 7 (Item Coverage ($\text{Cov}_{\%}$@K)). *We measure the percentage fraction of the number of different items in the top-K recommendation lists as follows*

$$\text{Cov}_{\%}\text{@K} = \frac{1}{|\mathcal{I}|} \sum_{u=1}^{\mathcal{I}} hit(i, K) \times 100 \tag{2.7}$$

where $hit(i, K)$ is if the item u has been recommended at least in one recommendation lists generated by the recommender, 0 otherwise. $\text{Cov}_{\%}\text{@K} = 100\%$ *means that the entire item catalog is covered by the recommender.*

2.2 Foundations of AML

Adversarial attack strategies have been firstly introduced in computer vision domain,

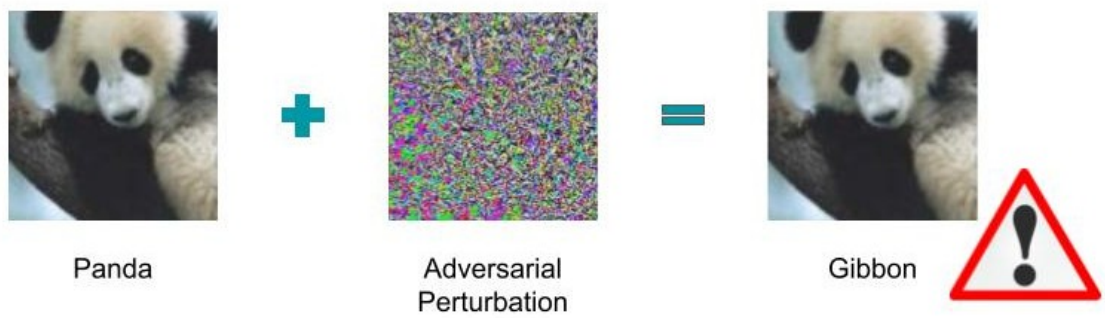


Fig. 2.1 Standard examples of the injection of adversarial perturbation to build an adversarial sample that lead a classifier to a wrong class prediction.

with a particular focus on image classification tasks. In a classical *supervised learning* setting, $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ denotes the dataset where $\mathbf{x}_i \in \mathcal{X} \subseteq R^d$ is a feature vector in the *input space* $\mathcal{X}$ and $y_i \in \mathcal{Y}$ is the corresponding label in the *output space* $\mathcal{Y}$. For instance, in binary classification $\mathcal{Y} = \{-1, +1\}$. Each pair in $\mathcal{D}$ is assumed to be independent and identically distributed (i.i.d) from an unknown distribution Φ, i.e., $(\mathbf{x}, y) \sim \Phi$. We also assume that we are given a suitable loss function $\mathcal{L}(.,.)$, 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

$$\min_{\Theta} \ \mathbb{E}_{(\mathbf{x},y)\sim\Phi} \ \mathcal{L}(f(\mathbf{x}; \Theta), y) \tag{2.8}$$

where $\mathbb{E}_{(\mathbf{x},y)\sim\Phi}$ is commonly termed *expected risk* of the classifier, Θ are the model parameters and y is the class label for the input sample $\mathbf{x}$. As Φ it is often unknown, we

use $\mathcal{D}$ in order to learn the suitable candidate function $f(\mathbf{x}, \Theta)$. The training objective function can be formulated as the following optimization problem,

$$\min_{\Theta} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}} \mathcal{L}(f(\mathbf{x}_i; \Theta), y_i) \tag{2.9}$$

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

However, while ERM is a powerful solution to train classifiers, it cannot learn models robust against adversarial images. In 2013 Szegedy et al. [208] found that, given an image, it is possible to add a meticulously crafted human-imperceptible perturbation such that a pre-trained deep neural network (DNN) will misclassify the adversarial samples. For example, as shown in Figure 2.1, an adversary may perturb pixels of a "pandas" image so that humans will not be able to observe changes, but the classifier produces "gibbon" as the classification result. Szegedy et al. [208] named the perturbed images as *adversarial examples* and presented the first adversarial strategy, known as L-BFGS, to learn the adversarial noise.

Before we dive into the applications of AML in RSs, we present its preliminaries and foundations in the computer vision domain, the pivotal field of AML studies.

2.2.1 Adversarial Attacks

Starting from the work by Szegedy et al. [208], an adversarial attack that aims to force a trained model to make a wrong prediction under a minimal perturbation budget can be defined as in Definition 8.

Definition 8 (Adversarial Perturbation)**.** *Given a learned classifier $f(\mathbf{x}; \Theta)$ and an instance from the dataset $(\mathbf{x}, y) \in \mathcal{D}$, the attacker takes the sample $\mathbf{x}$ and adds an adversarial perturbation δ to build the adversarial sample $\mathbf{x}_{adv} = \mathbf{x} + \delta$ such that $f(\mathbf{x}_{adv}; \Theta) \neq f(\mathbf{x}; \Theta)$. δ is defined as follows*

$$\max_{\delta} \ \mathcal{L}(f(\mathbf{x} + \delta; \Theta), y), \quad s.t., \ \|\delta\|_p \leq \epsilon, \tag{2.10}$$

*where ϵ is the **perturbation budget**, typically chosen as small as possible such that the p-norm of the perturbation ($||\delta||_p$) is below that limit.*

Equation (2.10) formally illustrates a fundamental aspect of adversarial attacks that generalizes over other domains, e.g., recommendation, that the perturbations are evaluated via a maximization (or minimization) problem with the characteristic to

be the smallest possible in order to find learning characteristics that destabilize the behavior of the model.

One of the first and most used adversarial methods to build adversarial samples in the fast gradient sign method (FGSM) proposed by Goodfellow et al. [101]. The FGSM attack model [101] was originally designed to exploit the linearity of DNNs in the higher dimensional space. The authors' goal is to solve Equation (2.10) (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 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.

Definition 9. *(Untargeted Fast Gradient Sign Method (FGSM)). The Untargeted Fast Gradient Sign Method is defined as follows*

$$\mathbf{x}_{adv} = \mathbf{x} + \epsilon \cdot sign(\bigtriangledown_x \mathcal{L}(f(\mathbf{x};\theta),\ y)) \tag{2.11}$$

where ϵ (perturbation level) represents the attack strength and $\bigtriangledown_x$ is the gradient of the loss function w.r.t. input sample $\mathbf{x}$, y is the correct label and $sign(\cdot)$ is the sign operator.

Definition 10 (Targeted Fast Gradient Sign Method (FGSM))**.** *The corresponding approach for targeted FSGM [142] is defined as follows*

$$\mathbf{x}_{adv} = \mathbf{x} - \epsilon \cdot sign(\bigtriangledown_x \mathcal{L}(f(\mathbf{x};\theta),\ y_T)) \tag{2.12}$$

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

Carlini and Wagner is another state-of-the-art attack model for finding adversarial perturbation under three distance measures (ℓ_0, ℓ_2, ℓ_∞). Its key insight is similar to L-BFGS [208] as it transforms the constrained optimization problem into an empirically chosen loss function to form an unconstrained optimization problem as

Definition 11 (Carlini and Wagner (C&W)). *Let $h(\cdot)$ be a candidate loss function (e.g., $f(\cdot)$), the C&W attack is formulated as*

$$\min_{\delta} \left(\|\delta\|_p^p + c \cdot h(\mathbf{x} + \delta, y_T) \right) \tag{2.13}$$

Then, since the C&W attack has been used with several norm-type constraints on perturbation (i.e., L_0, L_2, L_∞), the CW-ℓ_2 problem formulation for a targeted attack aiming is given by

$$\begin{aligned} &\min_{\delta} \left(\|\mathbf{x}_{adv} - \mathbf{x}\|_2^2 - c \cdot h(\mathbf{x}_{adv}, y_T) \right) \\ &h(\mathbf{x}_{adv}) = \max \left(\max_{i \neq t} \; Z\{\mathbf{x}_{adv_i}\} - Z\{\mathbf{x}_{adv_t}\}, -K \right) \\ &\mathbf{x}_{adv} = tanh(arctanh(\mathbf{x}) + \delta) + 1)) \end{aligned} \tag{2.14}$$

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

Before we dive into the presentation of background knowledge on defenses, it is worth mentioning the suggestion presented by Carlini et al. [56] in the context of research on security problems. In this work, the authors claim the necessity to define the adversary threat model to clearly outline what adversary's type a possible defense will intend to defend against, guiding the evaluation of the attack and the defense. The adversary threat model is based on assumptions about the goals, knowledge, capabilities, and time.

- The **adversarial's goal** consists of the malicious outcome that the adversary would like to obtain while building adversarial examples. For instance, the adversary's goal in the CV domain may be to cause misclassification. Then, any adversarial samples being misclassified is a successful attack.

- The **adversarial's capabilities** are defined to impose reasonable constraints to the attacker to allow defenses implementation that unconstrained adversaries do not trivially bypass. For example, adversarial defense defined to protect a specific class of attacks cannot be evaluated against another.

- The **adversarial's knowledge** clearly describes what knowledge the adversary is assumed to have concerning the model, the input data, and the output data. Typically, works assume either white-box, full knowledge of the attacked model, parameters, and data; black-box, no knowledge, and gray-box, a partial knowledge

of the model or the data. When designing a defense method, the guiding principle is to assume that the adversary has white-box knowledge such that the defense will be reasonably effective in black-box settings.

- The **adversarial's time** depends on the moment when the adversary performs the attack to change the behavior of the ML system. Adversarial timing can be at training or testing time. Training time attacks, also known as *poisoning attacks*, happen before the ML model is trained. The attacker can add false data points into the model training data, causing the trained model to produce an erroneous prediction [43]. Testing time attacks, also named as *evasion attacks*, aim to evade the decisions made by the learned model by maliciously manipulating the test samples [101].

2.2.2 Adversarial Defenses

From an all-inclusive view, defending an ML model against adversarial attack strategies can be done by (i) increasing the robustness of the learning or (ii) detecting the adversarial examples before the inference through the network.

Increasing the robustness of the learning algorithm consists of training strategies allowing the correct classification of adversarial and clean samples. The idea is to learn model parameters (Θ) less sensitive to minor data variations that might move samples into the wrong decision boundary. A standard strategy in the CV domain is to regularize models to mitigate the attack surface, learning to correctly classify the malicious samples. This problem can be formulated as a *robust optimization* problem that seeks to correctly classify the adversarial samples of a determined adversarial threat model.

Definition 12 (Robust Optimization)**.** *Let $\mathcal{L}$ be the loss function, f be the learning model characterized by the model parameters Θ, $(\mathbf{x}_i, y_i) \in \mathcal{D}$ be the training sample, $\|\delta\|_p \leq \epsilon$ the specification of the threat model, then the robust optimization is defined as follows*

$$\min_{\Theta} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}} \max_{\delta} \ \mathcal{L}(f(\mathbf{x}_i + \delta; \Theta), \ y_i) \tag{2.15}$$

Definition 13 (Adversarial Training)**.** *Based on the above minimax learning strategy, Goodfellow et al. [101] defined Adversarial Training as follows*

$$\min_{\Theta} \underbrace{\left[\mathcal{L}(f(\mathbf{x};\Theta),y) + \lambda \underbrace{\max_{\delta:\|\delta\|\leq\epsilon} \mathcal{L}(f(\mathbf{x}+\delta;\Theta),y)}_{\textit{Adversarial Regularizer}}\right]}_{\textit{Adversarial Regularized Loss}} \tag{2.16}$$

where $AdversarialRegularizedLoss$ is composed of two parts, the standard classification loss component ($\mathcal{L}(f(\mathbf{x};\Theta),y)$) plus $AdversarialRegularizer$, that is the loss evaluated on the adversarial samples (continually) created to violate the current model Θ. Finally, $\lambda \in \mathbb{R}$ is the adversarial regularization coefficient used to control the trade-off between accuracy (on clean data) and robustness (on perturbed data).

The following section presents how adversarial attacks and respective countermeasures have been adopted in the recommendation domain. When needed, further details of adversarial attacks and defenses will be specified in the related work of each chapter.

2.3 AML in Recommendation Task

Recommender models have been demonstrated to be steadily under security risks. Unlike a standard adversarial attack setting in CV where the adversaries perturb images such that a classifier makes wrong predictions, the setting in RS must be rethought according to input, model, and performance differences between classifiers and recommenders. In the next section, we identify and clarify the main differences before presenting a literature review of AML applications in RSs.

2.3.1 Differences Between RS and CV Settings.

As shown before in Section 2.2, a standard framework to assess the goal of an adversarial attack against a classifier is to build imperceptible perturbations that adversarially optimize to change the correct behavior of the model.

Input

The first fundamental difference lies in the type of model input. In a test time attack setting, the pixel-valued nature of image data makes feasible the injection of minimal noise (the adversarial perturbation) that slightly changes pixel values defeating the classifier accuracy (the model misclassifies the adversarial sample) while persisting

completely imperceptible for humans. Oppositely, the input data of a CF-RS is a pair of user and item identifiers (discrete values) whose variations completely change the semantic of the sample. For instance, suppose the adversary goal is to increase $\hat{s}(i \mid u)$, perturbing i would mean to change its ID from i to i', where $i' \neq i$. It follows that $\hat{s}(i' \mid u)$ is the score predicted on a different item, and the attack has no sense. Indeed, test time attacks are not feasible in pure collaborative models if not in case of adversarial perturbation on model parameters (Θ) that will be discussed in Section 2.3.2. While human-imperceptible test time attacks are not feasible on user-item preference data, it is feasible to create adversarial samples in the case of hybrid and content-based recommenders that make use of content data as described in Section 2.3.2.

Differently from the infeasibility in testing scenarios, training time attacks, even though with a partially different formalism, are executable in RSs. These can be performed as both creating fake profiles (or inserting/removing user-item feedbacks) and injecting adversarial perturbed content data— only in the case of hybrid and CBF RSs. Section 2.3.2 reviews the main research direction in the RS literature.

Model

Another aspect differentiating AML applications between the more popular classifiers in the CV domain and recommenders is the model type. In CV, standard image classifiers can be seen as a cascade of layers whose input, the image, is multiplied by the set of parameters related to the first layers, whose output will be the input of the second layers, and the process continues in this cascade until reaching the last year (the classification layer), that is the output. It follows that slight variations of the input will be propagated across the model $f(\Theta|\mathbf{x}_i) \neq f(\Theta|\mathbf{x}_i + \delta)$. In the case of CF-RSs, being two numerical identifiers inputs of the model, it will use them only to select the corresponding user and item rows in $\mathbf{P}$ and $\mathbf{Q}$. It means that it is not possible to model a test time attack against CF recommenders. It is the reason why the only testing time attacks investigated in the literature are related to adversarial perturbations of model parameters in-depth presented in Section 2.3.2.

Differently from CF, CBF and Hybrid recommenders are more suitable for testing time adversarial threat models. In this setting, assuming that the recommender model extracts the content feature at the moment of score prediction, it is possible to adversarially perturb the content to produce an altered output. For instance, the adversary can be a music creator that replaces the track of a song with an adversarial example in order to make the latent representation of the song closer to the most popular

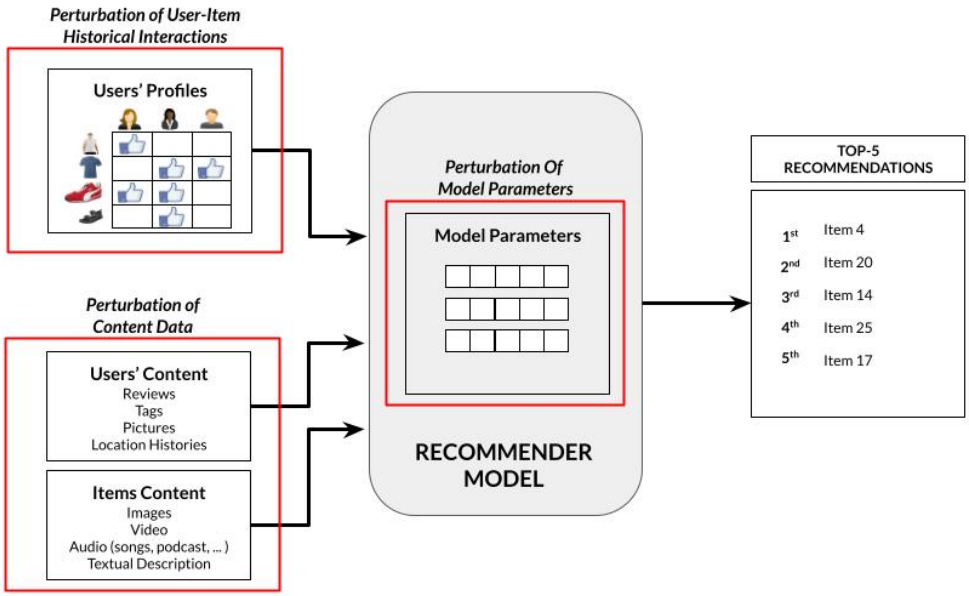


Fig. 2.2 A notional view of the possible injection of adversarial perturbations on (a) user profiles, (b) content data, and (c) model parameters.

ones and increase the frequency of recommendability in small top-K recommendation lists. Section 2.3.2 presents the state-of-the-art approaches in this setting.

Performance

Finally, evaluating the quality of a list of recommended items differs from evaluating the accuracy of a predicted class. Indeed, the adversary's goals in classification tasks are to lead the model to misclassify an adversarial sample with a chosen class (targeted attack) or any other one (untargeted attack) that is different from the original. In contrast, the goals in the recommendation task are different from the one in classification, such as to increase/decrease the predicted preference score, push/nuke the position in ranked lists, and make unreliable and not-personalized recommendations for a user or a group of users. This variety makes necessary the definition of complex adversary threat models.

2.3.2 Adversary Threat Models against RSs

In the current section, we classify the research areas on adversarial strategies in RSs and discuss the attack and defensive strategies according to the component, either the input of the model parameters, under the adversarial attack. The general schema for which parts of an RS can be under adversarial perturbations is shown in Figure 2.2.

Table 2.1 Different categories of AML applications in RSs (and example research in each case). We underline the fields where we produce a research contribution.

AML Applications	**References**
Perturbation of User-Item Interaction (Poisoning of $\mathcal{R}$)	
• Hand-engineered Poisoning	
∗ *Attack by leveraging interaction data*	[143, 174]
∗ *Attack by exploiting semantic data*	[16]
∗ *Studying the impact of data characteristics*	[76]
∗ *Defenses*	[42, 54, 244, 8, 50]
• Machine-learned Poisoning	
∗ *Factorization-based models*	[146, 65, 59, 93]
∗ *Reinforcement Learning models*	[239, 205, 55]
∗ *Other recommendation models*	[230, 62]
∗ *Defenses*	[153]
Perturbation of Content Data (Multimedia Recommenders)	
• Poisoning (Training Time)	
∗ *Targeted Adversarial Attacks*	[85, 27, 78]
∗ *Defenses*	[78]
• Evasion (Testing Time)	
∗ *Attacks on Scores and Rankings*	[67, 154, 23]
∗ *Defenses*	[154]
Perturbation of Model Parameters (Poisoning of Θ)	
• Embeddings of RSs	
∗ *Gradient-based attack: single-step, multi-step*	[115, 209, 12]
∗ *Gradient-based defenses*	[115, 229]
∗ *Performance trade-off with adversarial trained RSs*	[Under Review]

According to the adversarially perturbed element component of an RS shown in Figure 2.2, we can perform adversarial perturbations on the set of recorded preferences $\mathcal{R}$ (e.g., injection of fake interactions), the content data used as a side-information (e.g., the item images uploaded on an e-commerce fashion platform), and the model parameters (e.g., ideal attacks used to study the stability against a worst-case scenario). Before we dive into the analysis of these strategies, to provide an overview, Table 2.1 introduces the adversarial attacks, which have been used over the last few years in RS research. It highlights the reviewed research articles according to three dimensions: perturbation of user-item recorded interaction (poisoning of $\mathcal{R}$), perturbation of content data, and perturbation of model parameters (poisoning of Θ). In the following section, we describe each category.

Perturbation of User-Item Recorded Interaction (Shilling Attacks)

The rationale behind CF-RSs is to ease the customer navigation across the catalog based on the so-called "word-of-mouth," i.e., a user might like what other people like and dislike. However, the openness of these systems is a potential point of failure. Indeed, malicious users, the adversaries, can meticulously craft fake profiles to poison the data and alter the recommendation behavior toward malicious goals [173, 6, 42]. Adversaries may execute a **shilling attack** to achieve a whole different set of adversaries' goals. To name a few, they may want to demote competitor products [143], misuse the underlying recommendation system [105], or increase the recommendability of specific products [161, 85].

The adversary threat model to perform a shilling attack considers the adversary's knowledge to mount the attack, the adversary's goal, and the adversary's capability (i.e., the number of added profiles) [48, 202]. According to the adversary's knowledge, a shilling attack can be a *low-knowledge* or an *informed* attack. The former class indicates a limited amount of available data information accessible by the adversary [143, 163]. The latter class assumes a higher knowledge of dataset information, such as the rating distribution. In this case, the adversary might be able to craft more effective profiles [143, 173]. Additionally, the knowledge of the recommender model can be helpful to perform even stronger attacks [146]. Regarding the adversary's goal, the adversary might alter the recommender to *push* or *nuke* the recommendability of a product, or a class of products, named *target items*. Push attacks aim to increase the targeted item's appeal, while nuke attacks aim to lower their recommendation frequency. Also, the adversary's capability can depend on the number of fake profiles added to the system, a constraint on the number of modifications or a modification penalty, and what kinds of modifications are admissible (e.g., insertion only or arbitrary modification). A common approach to measuring the granularity of attack is to measure the percentage of added profiles over the total number of regular users in the systems [163, 80].

Additionally, we have identified two main techniques to perform the poisoning of recorded interactions: hand-engineered and machine-learned strategies. The following two paragraphs survey and present the principal works of each field.

Hand-engineered Attacks. In the beginning, the main focus of the research community on the security of RSs has been on *hand-engineered* shilling attacks against CF models where the intuition is to add fake user profiles whose general form is defined below.

Table 2.2 State-of-the-Art Hand-engineered Attack Strategies and Their Profiles Composition (*push* goal).

Attack Type	I_S		I_F		I_ϕ	I_T
	Items	Pref. Score	Items	Pref. Score		
Random [143]	$\emptyset$		$\frac{\sum_{u \in U} \lvert I_u \rvert}{\lvert U \rvert} - 1$	$rnd(N(\mu, \sigma^2))$	$I - I_F$	max
Love-Hate [163]	$\emptyset$		$\frac{\sum_{u \in U} \lvert I_u \rvert}{\lvert U \rvert} - 1$	min	$I - I_F$	max
Bandwagon [174]	$(\frac{\sum_{u \in U} \lvert I_u \rvert}{\lvert U \rvert})/2 - 1$	max	$(\frac{\sum_{u \in U} \lvert I_u \rvert}{\lvert U \rvert})/2$	$rnd(N(\mu, \sigma^2))$	$I - I_S - I_F$	max
Popular [175]	$\frac{\sum_{u \in U} \lvert I_u \rvert}{\lvert U \rvert} - 1$	min **if** $\mu_f < \mu$ **else** $min + 1$	$\emptyset$		$I - I_S$	max
Average [143]	$\emptyset$		$\frac{\sum_{u \in U} \lvert I_u \rvert}{\lvert U \rvert} - 1$	$rnd(N(\mu_f, \sigma_f^2))$	$I - I_F$	max
P. Knowledge [173]	$\frac{\sum_{u \in U} \lvert I_u \rvert}{\lvert U \rvert} - 1$	max	$\emptyset$		$I - I_S$	max

where (μ, σ) are the dataset average preference score and its variance, (μ_f, σ_f) are the filler item i_f rating average and variance, and min and max are the minimum and maximum preference score value. rnd function generates one integer (i.e., rating) from a discrete uniform distribution.

Definition 14 (Hand-engineered Shilling Profile ($\mathcal{SP}$))**.** *Let $\mathcal{I}_S$ denote the selected item set, $\mathcal{I}_F$ the filler set, $\mathcal{I}_\phi$ the unrated-item set, $\mathcal{I}_T$ the target item set, and given a Recommendation Problem, a* ***Shilling Profile*** *($\mathcal{SP}$) is defined as follows*

$$\mathcal{SP} = \mathcal{I}_S + \mathcal{I}_F + \mathcal{I}_\phi + \mathcal{I}_T \tag{2.17}$$

where $\mathcal{I}_S$ contains items identified by the attacker to exploit the owned knowledge to maximize the effectiveness of the attack, $\mathcal{I}_F$ holds randomly selected items for which rating scores are assigned to make the attack imperceptible. $\mathcal{I}_\phi$ includes items without ratings in the fake user profile, and $\mathcal{I}_T$ is the item is to push or nuke. The $\mathcal{SP}$ composition varies based on attack strategies.

Note that $\mathcal{I}_S$ and $\mathcal{I}_F$ are chosen depending on the attack strategy, and the attack size is the number of injected fake user profiles. Throughout the dissertation, we use $\phi = |\mathcal{I}_F|$ to represent the filler size, $\alpha = |\mathcal{I}_S|$ the selected item set size and $\chi = |\mathcal{I}_\emptyset|$ to show the size of unrated items. Table 2.2 summarizes the main parameters involved in the implementation of the most prominent shilling attacks against CF models.

In general, the literature explores two main challenges: proposing and investigating attack strategies with their effects on the recommendation performance [143, 174, 80, 76] and exploring defensive mechanisms [42, 54, 245, 244, 8, 50]. We refer to the recent survey by Si and Li [202] for major details on defense strategies.

Machine-learned Attacks. Starting from 2016, machine learning approaches have been emerged as techniques to build optimized shilling profiles [146]. In the literature, several methods have been proposed to perform machine-learned injection attacks,

characterized by the proposal of an optimization procedure to maximize the adversary's goal. Based on the observation that optimization methods are strictly related to the recommender model under attack, we classify the poisoning optimization methods based on targeted models: (i) factorization-based recommenders, (ii) reinforcement learning models, and (ii) other recommendation families.

In this first category, the first work to compute near-optimal data attacks for factorization-based recommendation models has been proposed by Li et al. [146] in 2016. The authors approximately compute gradients of the solution of an optimization problem based on first-order Karush-Kuhn-Tucker conditions to perform both integrity and availability attacks. Another research direction is given by the adoption of Wilcoxon-Mann-Whitney loss [34] to approximate the hit probability of finding the target item in the recommendation list (e.g., [94, 124, 93]). A research effort has been recently devoted to defensive strategies against this class of adversarial models. For instance, Hidano and Kiyomoto [120] propose a *trim matrix factorization algorithm*, a robust method integrating the trim learning, an approach that 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.

The reinforcement learning methods are characterized by the adversary's knowledge and capability, the state space, the action space, and the reward utility function. Unlike the previous methods, reinforcement-based attacks need only leverage the feedback from the RS instead of knowing and accessing the whole set of parameters (Θ) to learn the agent's policy. One of the first works, named LOKI [239], circumvent the time-consuming operation of retraining the victim recommender to get the feedback and update the attack strategy. The authors build *a local recommender simulator* to mimic the target model and make the reinforcement framework get reward feedback from the simulator under the assumption that adversarial samples generated for one of the recommenders could be used to attack the other. Another recent strategy, named PoisonRec [205], models the sequential attack behavior trajectory as a Markov Decision Process.

Although the factorization-based and the reinforcement learning-based data poisoning methods have driven the 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 Yang et al. [230], Fang et al. [94]. While, Chen et al. [62] present the first attempt to learn an optimal set of fake users for making worse **k-Nearest Neighborhood** models.

Perturbations of Content Data in Multimedia Settings

As discussed in Chapter 1, RSs can rely on additional side-information, such as images, audio, and track files, and a part of the contributions presented in the dissertation relies on image data in the Chapter 4. Indeed, in scenarios such as fashion, food, or point-of-interest recommendation, images associated with products have positively impacted the outcomes of consumption decisions, as images attract attention, stimulate emotion, and shape users' first impressions about products and brands. To extend the expressive power of RSs, visual-based recommender systems (VRSs) have recently merged that attempt to incorporate products' visual appearance of items [83]. Given the representational power of deep neural networks (DNNs) in capturing images' characteristics, state-of-the-art VRSs often integrate visual features extracted via a DNN — pre-trained, e.g., VBPR [114] and ACF [61], or learned end-to-end, e.g., DVBPR [134].

Even though this research field is relatively new in the recommender systems community, we have identified that the adversary's goals are mainly relative to minimally perturb the product images such that the single item (or a group of items) can increase the frequency of recommendation in the shortest top-K recommendation lists (e.g., $K = 10$). Additionally, similar to the CV domain, the adversary's capability is relative to perturbations limited inside the budget perturbation ϵ (e.g., $\epsilon \leq 32$). Interestingly, this AML application is the only one that allows preserving both adversary's timing classification in training and testing time attacks.

Poisoning (Training Time). Poisoning the training dataset with adversarial samples is a novel research topic with real-world applications in content-based or hybrid recommendation models. Imagine the following motivational example: a competitor is enthusiastic about increasing the recommendability of a category of products on an e-commerce platform, e.g., *sandals*, for economic profit. She can achieve this goal by just uploading adversarially perturbed product images of sandals that are misclassified by the DNN used in the VRS as a popular class of products, e.g., *running shoes*, allowing sandals to be pushed into the recommendation list of more users. This realistic attack scenario is deeply explored in Chapter 5 for the case of VRSs, where we present our research contributions [85, 27, 17]. However, it is still an open challenge to verify whether adversarial samples used to poison datasets used in other domains (e.g., music and video) can still be effective and, if it is the case, it needs further research for possible defenses.

Evasion (Testing Time). Recently, evading the model with adversarially perturbed content data, e.g., product images, has attracted attention with the proposal of novel adversarial strategies. Cohen et al. [67] propose three attack strategies to push a target item to higher positions. Inspired by the fast gradient sign method by Goodfellow et al. [101], the first one is an iterative white-box strategy defined as below.

Definition 15 (White-box Sign-based Attack (WB-Sign))**.** *Let* $\mathbf{x}_i$ *be the image associated to the item* $i \in \mathcal{I}$*, let* $t \in \{0, 1, ..., T-1\}$ *where* $T \in \mathrm{N}$ *defines the number of attack iterations, then the adversarial sample is computed as*

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \epsilon \cdot \mathrm{sign}\left(\frac{\partial s}{\partial \mathbf{x}_i^t}\right) \tag{2.18}$$

where sign *is* $+1$ *when the gradient is positive, otherwise* -1*, and si is the preference score function applied an all the users in the system.*

Additionally, the authors proposed two black-box strategies, named Black-Box Attack on Scores (BB-Score) and Black-Box Attack on Rankings (BB-Rank), proposing an approach for numerical computation of the partial derivatives of unknown recommender model function $s\cdot$.

Additionally, Liu and Larson [154] propose three attacks with different levels of adversary's knowledge. Similar to Definition 15, the white-box attack assumes that the adversary knows the model parameters and can build the perturbation by maximizing the score produced in that product image. The middle-knowledge attack assumes that the adversary knows the used DNN to extract the image features and which are the most popular products in the catalog (named *hook items*). She uses this knowledge to build perturbation such that the feature extracted from the target image is as close to the one of a very most popular product. Finally, in the limited-knowledge setting, the adversary slightly modifies the image, adding a visual component of popular products (e.g., add a pair of popular pairs of shoes in the image of a jeans' product). Novel test time attack methods have been also proposed by Cohen et al. [67]. Further formalization details will be presented in Chapter 6 that is completely focused on this class of malicious strategies.

Perturbations of Model Parameters

The third class of adversarial methods proposed in recommendation scenarios is related to applying adversarial perturbations on model parameters (Θ) to verify their

stability in a worst-case adversarial context (i.e., the adversary knows the model parameters and the learning procedure and can access them).

Since model parameters are vectors with continuous values, a part of the theory and methodologies common to build perturbations for breaking image classifiers (see Section 2.2) have been re-adapted in the case of model-based RSs. For instance, using the matrix factorization (MF) model trained with BPR (known as BPR-MF) — the state-of-the-art ranking-based criterion for item recommendation — He et al. [115] have investigated the robustness of embedding parameters when FGSM-based perturbations are added to user embeddings (i.e., $\mathbf{P}+\delta$) and item embeddings (i.e., $\mathbf{Q}+\delta$).

Attack Methods. He et al. [115] have studied the robustness of BPR-MF [188] proposing adapting the FGSM approach by linearizing the recommender loss function $\mathcal{L}$ around an initial zero-matrix perturbation δ_0 and applying the max-norm constraint.

Definition 16 (FGSM-based Perturbation on Model Parameters Θ)**.** *The adversarial perturbation δ^{adv} is defined as:*

$$\delta^{adv} = \epsilon \frac{\Pi}{\|\Pi\|} \quad \textit{where} \quad \Pi = \frac{\partial \mathcal{L}(\Theta + \delta_0)}{\partial \delta_0} \tag{2.19}$$

where $||\cdot||$ is the L_2-norm.

After the calculation of δ^{adv}, He et al. [115] have added this perturbation to the current model parameters $\Theta^{adv} = \Theta + \delta^{adv}$ and generated the recommendation lists with this perturbed model parameter to demonstrate that the noise with $\epsilon = 0.5$ would have impaired the recommendation accuracy 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 collaborative auto-encoders [235, 234], visual-based recommender [209], tensor-factorization [58], sequential recommendations [156], and attentive song recommenders Tran et al. [211].

Another adversarial strategy inspired by the CV domain is the Carlini & Wagner (C&W) attack [57]. Indeed, Du et al. [89] have shown how it may contaminate the model performance in the testing phase adapt the C&W approach to a recommender model (i.e., neural collaborative filtering [116]). The C&W optimization problem is formulated as follows:

Definition 17 (C&W-based Perturbation on Model Parameters Θ)**.** *Let $p(\cdot)$ be the prediction function to mark an item relevant to a user*

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

The authors demonstrated that the attacks got a success rate close to 100% in inverting the predicted importance ($p(\cdot)$) of each user-item pairs.

Defense Strategies. The above-presented perturbation strategies against model-based recommenders make evident their vulnerability to little noise on model parameters. An adversary may access the model and completely misuse a recommender's utility by slightly perturbing their learned parameters. 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. Authors [115] 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 tremendous negative change in performance.

The identification of such instability have raised the need of proposing defense strategies. Inspired by the robust optimization mechanism (see Section 2.2), He et al. [115] proposed the first method that modifies the *BPR*-based loss function of an MF model implementing RS-oriented adversarial training procedure, named adversarial personalized ranking (APR), and defined as follows

Definition 18 (Adversarial Personalized Ranking (APR))**.** *APR learns Θ within the minimax optimization game*

$$\arg\min_{\Theta} \max_{\delta_{adv}, \|\delta_{adv}\| \leq \epsilon} \underbrace{\mathcal{L}_{BPR}(\Theta) + \alpha \mathcal{L}_{BPR}(\Theta + \delta_{adv})}_{:= \mathcal{L}_{APR}(\Theta)} \tag{2.20}$$

where $\mathcal{L}_{APR}(\Theta)$, the APR objective function, is composed by the standard BPR loss, i.e., $\mathcal{L}_{BPR}$, and a regularization term, i.e., $\mathcal{L}_{BPR}(\Theta + \delta_{adv})$, whose strength is controlled by α, named adversarial regularization coefficient.

This additional regularization term, named adversarial regularizer, is the loss obtained when an adversarial perturbation δ_{adv} is added to Θ to **maximize** the

model objective. It follows that, being δ_{adv} fixed, APR **minimizes** both the standard BPR loss $\mathcal{L}_{BPR}$ with, and without, δ_{adv}. APR aims to learn a model that can correctly distinguish the positive and negative items in the case of adversarial perturbations. Note that major details of the BPR optimization framework will be presented in Chapter 8.

APR inspired a series of robustness studies on other recommendation tasks. For instance, Tang et al. [209] applied the vulnerability study and proposed the APR defense to a visual-based RS for fashion recommendation. Yuan et al. [235, 234] investigated the robustification benefits of APR on a class of deep learning recommenders, the collaborative auto-encoder. Chen and Li [58] adopted the same approach to tensor-factorization models. Tran et al. [211] used APR for automatic playlist continuation. Manotumruksa and Yilmaz [156] implemented APR on a self-attention sequential recommender.

Additionally, [89] proposed by a form of **defensive distillation** [178] to make a deep recommender model (i.e., NeuMF) more robust to the C&W attacks presented in Definition 17. They distill the knowledge learned from a teacher model into a student (architecturally smaller) model. For instance, the items and users' latent vectors can be distilled into two lower-dimensional latent vectors. Furthermore, the authors have integrated the student model with a *noisy layer* for increasing the robustness of parameters against the perturbations. In the end, this procedure has been demonstrated to reduce the success rate of the C&W attacks compared to the baseline version of the recommender.

It is essential to mention that several defense strategies, as well as hundreds of adversarial attack strategies, have been designed and implemented in various domains (e.g., computer vision, speech recognition, and test processing) [227], and only a few of them have been already adapted in recommendation tasks. In Chapter 7, we will present a novel method contribution regarding this set of AML applications in RSs.

2.3.3 Evaluation Protocol

This last section is devoted to the analysis of the methodologies to evaluate AML application in RSs. After having identified the adversary threat model as specified in Section 2.2.1, to evaluate the efficacy of attack techniques, it is essential to define what is the *clean* setting from which we expect the adversary is trying to modify the standard behavior towards a malicious one. Then, the evaluation of defense strategies has to be led considering the (clean, under-attack, defended) triplet of adversarial settings.

Evaluation of Model Availability

In general, the evaluation protocol depends on the adversary's goal. In settings where adversaries try to break the model's availability (e.g., reduce the model accuracy), the standard evaluation approach measures the percentage change in recommendation metrics. For instance, He et al. [115] have measured the percentage reduction of nDCG@K) and HR@K), two popular ranking-based accuracy metrics presented in Section 2.1.

Evaluation of Model Integrity

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. In the case of poisoning attacks on the user-item interaction data with the adversary's goal to push/nuke and item or segment of items, the evaluation metric can be classified according to the *prediction accuracy* and the *stability*. Recommendation accuracy measures if the actual rating predicated by the recommendation model was altered due to the attack. Recommendation stability measures if the recommendation model recommends different products due to the attack irrespective of their actual preference score value [173].

Definition 19 (Hit Ratio on Target Items (HR@K($\mathcal{I}_T$,$\mathcal{U}_T$))). *Let $\mathcal{I}_T \subseteq \mathcal{I}$ be the set of target items under attack, let $\mathcal{U}_T \subseteq \mathcal{U}$ be the set of users under evaluation, then the Hit Ratio on Target Items* HR@K($\mathcal{I}_T$,$\mathcal{U}_T$) *is defined as follows*

$$\text{HR@K}(\mathcal{I}_T,\mathcal{U}_T) = \frac{\sum_{i \in \mathcal{I}_T} hit(i,\mathcal{U}_T)}{|\mathcal{I}_T|} \tag{2.21}$$

where $hit(i_t,\mathcal{U}_T)$ is the fraction of users in $\mathcal{U}_T$ for which item $i \in \mathcal{I}_T$ is ranked in the top-K recommendation lists [7].

Definition 20 (Prediction Shift on Target Items (PS($\mathcal{I}_T$,$\mathcal{U}_T$))). *Let $\mathcal{I}_T \subseteq \mathcal{I}$ be the set of target items under attack, let $\mathcal{U}_T \subseteq \mathcal{U}$ be the set of users under evaluation, then the Prediction Shift on Target Items* PS($\mathcal{I}_T$,$\mathcal{U}_T$) *is defined as follows*

$$\text{PS}(\mathcal{I}_T,\mathcal{U}_T) = \frac{\sum_{i \subset \mathcal{I}_T, u \subset \mathcal{U}_T} (\hat{s}_{ui} - s_{ui})}{|\mathcal{I}_T| \times |\mathcal{U}_T|} \tag{2.22}$$

This metric, originally proposed for shilling attacks, can be adopted in other adversarial settings. In Chapter 5, we will present an extension of HR@K($\mathcal{I}_T$,$\mathcal{U}_T$) in the case of perturbation on content data, and an extension for the case of nDCG@K.

This concludes the presentation of the background knowledge and related works required to comprehend the research contributions presented in the following chapters of the dissertation. When needed, each chapter will also include its own review of chapter-specific related works as needed.

2.4 Table of Abbreviations and Symbols

Table 2.3 Table of abbreviations used in this dissertation.

Abbreviation.	**Name**
ML	Machine Learning
AML	Adversarial Machine Learning
RS	Recommender System
CF	Collaborative Filtering
CBF	Content-based Filtering
MRS	Multimedia Recommender System
VRS	Visual-based Recommender System
KG	Knowledge Graph
UIFM	User-Item Feedback Matrix
DNN	Deep Neural Network
IFE	Image Feature Extractor
GAN	Generative Adversarial Network
MF	Matrix Factorization
LFM	Latent Factor Model
NeuMF	Neural Matrix Factorization
FGSM	Fast Gradient Sign Method
PGD	Projected Gradient Descent
C&W	Carlini and Wagner
AT	Adversarial Training
FAT	Free Adversarial Training
BPR	Bayesian Personalized Ranking
APR	Adversarial Personalized Ranking
EF	Explanatory Framework

s

Table 2.4 Table of Symbols used in this dissertation.

Symbol	**Description**
$\mathcal{U}$	Set of Users
$\mathcal{I}$	Set of Items
$\mathcal{R}$	Set of (u, i) Pairs
s_{ui}	Preference Score of User u on Item i
$\hat{s}(\cdot)$	Predicted Preference Score Function of
top-K	First K items sorted by $\hat{s}(\cdot)$
$\mathbf{P} \in \mathbb{R}^{\lvert\mathcal{U}\rvert \times h}$	Users' Embedding Matrix
$\mathbf{Q} \in \mathbb{R}^{\lvert\mathcal{I}\rvert \times h}$	Items' Embedding Matrix
$\mathbf{p} \in \mathbf{P}$	User' Embedding Vector
$\mathbf{q} \in \mathbf{Q}$	User' Embedding Vector
b_u	Observed User u Bias
b_i	Observed Item i Bias
$\mathbf{x}$	Input Vector of a Neural Network
y	Output Class/Label
Θ	Model Parameters
$\mathcal{SP}$	Shilling Profile
$\text{sign}(\cdot)$	Sign Operator
$f(\cdot)$	Inference Function of a Neural Network
$\mathcal{L}(\cdot)$	Loss Function
δ	Adversarial Perturbation/Noise
ϵ	Perturbation Budget
α	Adversarial Regularization Coefficient

Chapter 3

Impact of Data Characteristics on the Recommendation Robustness

Is there an underlying relationship between the dataset characteristics computed on the matrix of recorded user-item interactions and the effectiveness of shilling attack against collaborative recommender models?

Shilling attacks against collaborative filtering models consist of fake user profiles injected into the system by an adversary with the goal to harvest recommendation outcomes toward an evil desire. The source of CF's vulnerability is in the learning reliance on the user-item interaction data— like user-item ratings — to train their models and their inherent inability to distinguish genuine profiles from non-genuine ones. The majority of works conducted to analyze shilling attacks primarily focused on properties such as confronted recommendation models, recommendation outputs, and even users under attack. However, the under-researched element has been the impact of data characteristics on the effectiveness of shilling attacks against CF-RSs. Toward this goal, this chapter presents a systematic and in-depth study by using an analytical modeling approach built on a regression model to test the hypothesis of whether dataset properties can impact the robustness of CF recommenders under attack. Extensive experiments involving 97200 simulations on three different domains show that dataset properties affect the robustness of CF models. The results can help the system designer understand the cause of variations in RS performance due to a shilling attack.

3.1 Introduction

CF plays a pivotal role in online services in increasing traffic and promoting sales. This technique is widely adopted by various e-commerce and consumer-oriented services to recommend a whole range of items (e.g., products, music, and movies) with many real-world successful applications [99, 204]. As shown in Section 2.3.2, notwithstanding their great achievement, CF models are vulnerable to shilling attacks due to their open nature and inability to distinguish genuine user profiles from fake ones. For instance, Jannach et al. [131], Alonso et al. [9] have shown that a surprisingly modest number of fake profile attacks (around 3%) mounted on CF models are sufficient to manipulate a prediction shift up to 30%, signifying the impact that such handcrafted attack profiles can have on faulting recommendation results. As CF models assist users in many decision-making and mission-critical tasks, such non-robust measures could have far-reaching consequences, impacting peoples' lives and leaving the usability of RS questionable.

While existing works have focused on the design of attack and defense strategies, a common characteristic of them is that the experimental evaluation orientates to "*win-lose*" predicting scenarios, trying to find an answer to questions such as *Which attack models impact more the performance of specific recommendation models?* "*Which amount of knowledge on a specific recommendation model is required for specific attack A to influence recommendation algorithm B?*". Little effort has been made to provide an explanatory study on which dataset characteristics impact the effectiveness of attacks. For instance, it is known that RS performance can be affected based on the *sparsity of the dataset*, meaning that a highly dense dataset can impact the quality of CF models in ways that are different from a highly sparse dataset. However, whether this data characteristic can have a similar impact on the effectiveness of the profile injection attack remains far more under-researched.

In this chapter, we put our attention outside the subject of proposing another attack strategy against the recommendation model. Instead, we focus on the central question "*Given popular shilling attack types and CF models already recognized by the community, which dataset characteristics can explain an observed change in the performance of recommendation*?" This question is inspired by the work done by Adomavicius and Zhang [5] which studies the influence of rating data characteristics on the recommendation performance of popular collaborative RS. However, their work differs from ours because we utilize the explanatory model to explain the variation in the robustness of CF models (or effectiveness of attack strategies) concerning data characteristics.

We present a systematic and in-depth study of the impact of dataset characteristics on the robustness of CF by utilizing a regression-based model. Through a large-scale experiment on three domains, we evaluate how six data characteristics may influence the robustness of CF algorithms measured in terms of stability metrics. The proposed approach and the empirical evaluation carefully consider key contributions:

1. **Modeling**: we present a systematic, in-depth exploratory research and analysis of the impact of dataset characteristics on the robustness performance of popular CF models subjected to famous shilling attack strategies. To investigate the relationship between data characteristics and the robustness of CF models, we use regression-based explanatory modeling.

2. **Data characteristics**: unlike prior works on shilling attacks [163, 9], we validate the correlation between data characteristics and attack effectiveness on a suite of data characteristics extracted from the user-rating matrix ($\mathcal{R}$), going beyond well-recognized properties such as data sparsity.

Through extensive experiments, we analyze the regression model on six popular attack strategies against three well-known CF models across three real-world datasets. 97,200 attack simulations are conducted to solve the coefficient related to different explanatory regression problems (see Section 3.3). We rely on a statistical significance test with informed *p-value* to validate the hypothesis if the demonstrated set of data characteristics have an impact on the final model output.

3.2 Method

In this section, we describe the explanatory framework proposed to investigate the impact of data characteristics on attacks' effectiveness.

3.2.1 Independent Variables (IV)

This chapter focuses only on rating-based CF models as recommendation models exposed to shilling attacks. CF uses only the $\mathcal{R}$ to compute recommendations. For this reason, all the IVs representing dataset characteristics presented in this chapter are related to $\mathcal{R}$ characteristics and are inspired from [5]. We categorize these features according to (i) structure of $\mathcal{R}$, (ii) rating frequency of $\mathcal{R}$ and, (iii) rating values of $\mathcal{R}$.

IVs based on the $\mathcal{R}$ structure

The IVs that describe the structure of $\mathcal{R}$ are $SpaceSize_{log}$, $Shape_{log}$, and $Density_{log}$. Computing these IVs requires knowledge about the number of real users $|\mathcal{U}|$, the number of real items $|\mathcal{I}|$, and the number of recorded ratings $|\mathcal{R}|$.

Definition 21 ($SpaceSize_{log}$)**.** *Given a Recommendation Problem, we define $SpaceSize_{log}$ as in the following.*

$$x_1 = \log_{10}(\frac{|\mathcal{U}| \cdot |\mathcal{I}|)}{sc}) \tag{3.1}$$

The scaling factor (sc) is a parameter that can be set to limit the range of $|\mathcal{U}| \cdot |\mathcal{I}|$ into a small range. The $\log_{10}$ operation normalizes the distribution of this variable.

$SpaceSize_{log}$ is a variant of the original $SpaceSize_{log}$, and it is introduced in [5]. It is noteworthy that it may affect the performance of the underlying CF model and the mounted shilling profiles. For instance, under comparable density values, higher $\mathcal{R}$ $SpaceSize_{log}$ might imply a bigger chance of finding similar neighbor users or items. Therefore, as both attack and recommendation models rely on the identification of like-minded users (neighbor users) or similarly rated items (neighbor items), we deem $\mathcal{R}$ $SpaceSize_{log}$ to be an impactful dataset characteristic on evaluating the performance of shilling attacks applied on CF models. For instance, for the small dataset generated in this chapter during the simulations, typical values were in the range of *thousands* to *millions* with a wide variety. All of this can impact the accuracy of the regression model's coefficients calculated. Similar to [5], we set $sc = 1000$ in this work.

Definition 22 ($Shape_{log}$)**.** *Given a Recommendation Problem, we define $Shape_{log}$ as follows*

$$x_2 = \log_{10}(\frac{|\mathcal{U}|}{|\mathcal{I}|}) \tag{3.2}$$

$Shape_{log}$ can impact the effectiveness of shilling profile injection attacks. For example, in domains where the $Shape(\mathcal{R}) << 1$ (i.e., $|\mathcal{U}| << |\mathcal{I}|$) there are more candidate neighbor users than candidate neighbor items for memory-based CF models. This situation might work to the advantage of user-based CF than item-based CF. Moreover, under a similar number of ratings, changing the shape implies changing the average number of ratings per item $|\mathcal{R}|/|\mathcal{I}|$. We conjecture that this circumstance may impact the performance of CF under attacks, since the construction of the shilling profile is mainly based on altering the popularity of targeted items. The logarithm transformation in $Shape_{log}$ is applied to normalize the $|\mathcal{U}|/|\mathcal{I}|$ distribution. For example, the minimum and maximum values of shape in the MovieLens dataset are 0.366 and

30.039 before the log transformation, while -0.437 and 1.478 after the application of this operation.

Definition 23 ($Density_{log}$)**.** *Given a Recommendation Problem, we define* $Density_{log}$ *as in the following.*

$$x_3 = \log_{10}(\frac{|\mathcal{R}|}{|\mathcal{U}| \times |\mathcal{I}|}) \tag{3.3}$$

Data sparsity, which relates to data density, according to $density = 1 - sparsity$ is a well-recognized issue in the community of RS [1]. Data sparsity refers to situations where the fraction of unrated items significantly exceeds the fraction of rated ones, and not sufficient information is available for CF models to be trained and make predictions. Data sparsity can harm the performance of CF in different ways. For instance, it can reduce the chance of discovering neighbors in memory-based CF because the possibility of having co-rated items is lower in sparse $\mathcal{R}$. Model-based CF can suffer significantly from the sparsity problem to train [70]. A large amount of research focuses on investigating and alleviating the sparsity problem in CF recommender systems by proposing various solutions [126, 49, 117]. In [80], the authors identify a potential impact of dataset density on the effectiveness of shilling attacks.

IVs based on the $\mathcal{R}$ rating frequency

Another analyzed characteristic of $\mathcal{R}$ is the rating frequency distribution. The idea is that in many real applications, a few items receive numerous ratings (short heads or popular items), while a large number receive low or few feedbacks (long tails), causing the rating distribution to be skewed. It turns out that the commercial profit from recommending long-tail items is more significant than short-head items [158]. However, these long-tail items have less chance to be recommended since they have less historical feedbacks [168]. We examine this $\mathcal{R}$ characteristic because in a very skewed situation (e.g., few items with many ratings), the possibility to alter recommendations could be very low because popular items will be recommended by themselves.

Definition 24 ($Gini_{item}$, $Gini_{user}$)**.** *Given a Recommendation Problem, let* $|\mathcal{R}_i|$ *be to the number of ratings received by the item* i*, let* $|\mathcal{R}_u|$ *be to the number of ratings given by the user* u*, we define* $Gini_{item}$ *and* $Gini_{user}$ *respectively as in the following:*

$$x_4 = 1 - 2\sum_{i=1}^{|\mathcal{I}|}(\frac{|\mathcal{I}| + 1 - i}{|\mathcal{I}| + 1}) \times (\frac{|\mathcal{R}_i|}{|\mathcal{R}|}) \tag{3.4}$$

[1] We describe data sparsity since it is a more common term in the literature of RS, but everything mentioned on the sparsity relates to density in a reverse manner.

$$x_5 = 1 - 2\sum_{u=1}^{|\mathcal{U}|}(\frac{|\mathcal{U}|+1-u}{|\mathcal{U}|+1}) \times (\frac{|\mathcal{R}_u|}{|\mathcal{R}|}) \tag{3.5}$$

We use the Gini coefficient that measures the concentration of items, or users', ratings to capture the rating frequency distribution. The equal popularity (e.g., all users give the same number of ratings) is represented with the value of the Gini coefficients to 0, while the total inequality (e.g., only one user has given all ratings) is represented with the value to 1.

IVs based on rating values of the $\mathcal{R}$

While the previous dataset characteristics relate to the structure of the $\mathcal{R}$ and the distribution of ratings assigned to items, they disregard the actual values of the ratings themselves. The most common statistics representing rating values are *rating mean* and *rating variance*. Similar to [5], we disregard the overall rating means because most CF models involve a pre-processing step that centralizes the rating around the mean rating value, effectively removing its effect. Therefore, we study the effect of rating variance by investigating the following measure.

Definition 25 (Std_{rating})**.** *Given a Recommendation Problem, let $\bar{s}$ bet the global mean value of the scores (i.e., ratings) in the $\mathcal{R}$, we define Std_{rating} as in the following.*

$$x_6 = \sqrt{\frac{\sum_{i=1}^{|\mathcal{R}|}(s_i - \bar{s})^2}{|\mathcal{R}|-1}} \tag{3.6}$$

We investigate the possible influence of Std_{rating} on the robustness analysis motivated by the connection between high rating variance and recommendation performance claimed by Herlocker et al. [119] and the linear and negative impact on the accuracy performance reported in [5].

3.2.2 Dependent Variables (DV)

The dependent variable (DV) measures the effectiveness of the attack on RS. To this purpose we define the Incremental Overall Hit Ratio at K as follows

Definition 26 (Incremental Overall Hit Ratio at K ($\Delta_{\mathrm{H}}\mathrm{R}(\mathrm{U_T},\mathrm{I_T})@\mathrm{K}$))**.** *Let* $\mathrm{HR@K}(\mathcal{I}_T,\mathcal{U}_T)$ *be the metric value before the attack,* $\hat{\mathrm{HR}}@\mathrm{K}(\mathcal{I}_T,\mathcal{U}_T)$ *the value after an attack, the Incremental Overall Hit Ratio is defined as*

$$\Delta_{HR@K}(\mathcal{I}_T,\mathcal{U}_T) = \hat{\mathrm{HR}}@\mathrm{K}(\mathcal{I}_T,\mathcal{U}_T) - \mathrm{HR@K}(\mathcal{I}_T,\mathcal{U}_T) \tag{3.7}$$

where higher values mean more powerful attack in push cases, worse attack in nuke ones.

Evaluation metric for shilling attack effectiveness can be classified according to: *prediction accuracy* and *stability*. Recommendation accuracy measures if the actual rating predicated by the recommendation model was altered due to the attack. Recommendation stability measures if the recommendation model recommends different products due to the attack irrespective of their actual rating value [173]. The Incremental Overall Hit Ratio is a stability metric introduced for the explanatory modeling analysis.

3.2.3 Explanatory Framework (EF)

Statistical models can be used for two purposes: (i) explanatory modeling (EM) and (ii) predictive modeling. EM seeks to test the causal hypothesis into a theoretical construct, which means if a set of underlying effects measured by $\mathbf{X}$ are the cause for an underlying effect measured by y. The goal of predictive modeling, on the other hand, is to predict new or future observations given their input values ($\mathbf{X}$) [201]. Furthermore, (i), the model is carefully constructed to support the interpretability of the relationship between $\mathbf{X}$ and y, while in (ii) the model is "constructed from data". Prior works on shilling attacks have been largely focused on predictive approaches to improve the performance of attacks [105, 8, 173]. Instead, in this work, we choose a different approach and adopt an EM approach to test the causal hypothesis between underlying factors representing data characteristics ($\mathbf{X}$) and the underlying effect represented by attack performance (y). Grounded on [5], we use a formal method based on the regression model as a classical interpretable EM function.

Given a dataset d, a shilling attack strategy, a CF recommendation model g (e.g., item-based CF, user-based, and MF), then the goal is to test the hypothesis whether the factors related to dataset characteristics measured by $\mathbf{X}$ (IVs) can explain the effect on the RS performance measured by y (DV). In our settings, the dependent variable is represented by a metric able to measure the effects of a shilling attack. A regression model is used to model the causal relationship in the presented framework

$$y_i = \epsilon_i + \theta_0 + \sum_{d=1}^{D-1} \theta_d x_{d,i} + \sum_{c=1}^{C} \theta_c x_{c,i} \tag{3.8}$$

in which C is the number of data characteristic factors, θ_c is the regression coefficient of the c-th IV and $x_{c,i} \in \mathbb{R}$ represents the value of the c-th independent variable for the

i-th training example, and $y_i \in \mathbb{R}$ is the measurement corresponding to i-th training example (the measured dependent variable). $\sum_{d=1}^{D-1} \theta_d x_{d,i}$ is a dummy term introduced only for the between-datasets analysis (cf. Section 3.3.2), whose role is to capture information about dataset variation, where D is the number of the datasets in the across datasets study, $x_{d,i}$ is a binary $(0,1)$ dummy variable representing whether sample i belongs to the dataset d or not, and θ_d is the regression coefficient associated with the dataset d.

In a more compact way, we have

$$\boldsymbol{y} = \boldsymbol{\epsilon} + \theta_0 + \boldsymbol{\theta}_d \mathbf{X}_d + \boldsymbol{\theta}_c \mathbf{X}_c \tag{3.9}$$

where under mean-centered data, θ_0 represents the expected value of $\boldsymbol{y}$ (the performance metric under analysis), $\boldsymbol{\theta}_d = [\theta_1, \theta_2, ..., \theta_{D-1}]$ is the vector containing coefficients of the dummy variable $\mathbf{X}_d$ related to the dataset of each training example, $\boldsymbol{\theta}_c = [\theta_1, \theta_2, ..., \theta_C]$ is the vector of the regression coefficient associated with the IVs, and $\mathbf{X}_c$ is the matrix containing the independent variables values (data characteristic values computed based on $\mathcal{R}$).

We apply the regression framework to address two explanatory analyses: (i) within-dataset and (ii) between-dataset analyses presented in the following paragraphs.

Within-dataset analysis. The within-dataset analysis addresses the task of analyzing the impact of $\mathcal{R}$ characteristics for each combination of datasets, type of attacks, and recommendation models. The regression coefficients in the linear explanatory model are computed under the ordinary least squares (OLS) optimization model. The OLS minimization problem is defined as follows:

$$(\theta_0^*, \boldsymbol{\theta}_c^*) = \min_{\theta_0, \boldsymbol{\theta}_c} \frac{1}{2} \|\boldsymbol{y} - \theta_0 - \boldsymbol{\theta}_c \mathbf{X}_c\|_2^2 \tag{3.10}$$

Section 3.3.2 analyzes the regression model results for the within-dataset analysis.

Between-dataset analysis. We extend the within-dataset analysis explanatory model to a between-dataset analysis with the goal to examine a domain-independent perspective about the impact of data characteristics on the model output. The minimization problem is defined as follows:

$$(\theta_0^*, \boldsymbol{\theta}_d^*, \boldsymbol{\theta}_c^*) = \min_{\theta_0, \boldsymbol{\theta}_d, \boldsymbol{\theta}_c} \frac{1}{2} \|\boldsymbol{y} - \theta_0 - \boldsymbol{\theta}_d \mathbf{X}_d - \boldsymbol{\theta}_c \mathbf{X}_c\|_2^2 \tag{3.11}$$

Table 3.1 The dataset statistics related to the dataset used in this work.

Dataset	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{R}\rvert$	*density*
ML-20M	138,493	26,744	20,000,263	0.0054
Yelp	25,677	25,778	705,994	0.0010
LFM-1b	120,175	521,232	25,285,767	0.0004

where the constant term θ_0 represents the reference dataset (in our experimental evaluation we consider ML-20M) and the dummy term $\boldsymbol{\theta}_d\mathbf{X}_d$ provides a binding to the other $D-1$ datasets (i.e., Yelp and LFM-1b in our experiments) [5]. Section 3.3.2 presents the regression model results for the between-dataset analysis.

3.3 Experiments

In this section, we present experimental settings and a discussion of the results.

3.3.1 Settings

Datasets

We conducted shilling attacks against CF models on three real-world datasets, ML-20M [109], Yelp [115], and LFM-1b [195]. The datasets have properties that are considerably different from each other — for instance, considering the domains and structural properties of the dataset (see Table 3.1)—, effectively allowing us to analyze and validate the experimental results under a diverse set of data characteristics.

- **ML-20M** [109] is a 20 million-sized version of MovieLens (ML) dataset. Each item (movie) is rated on 0-5 Likert scales. ML is among the most commonly adopted datasets for the offline evaluation of RS, and ML-20M is the largest stable version among different dataset variations.

- **Yelp** [115] contains users' ratings, reviews, and check-in on businesses (e.g., restaurants) collected from Yelp.com. We used a pre-processed version of the dataset provided by [115] that contains only integer rating values in the range $(1,5)$ assigned by users to businesses.

- **LFM-1b** [195] is a music domain dataset containing more than one billion listening events (e.g., playing a track of an artist) fetched from January 2013 to August 2014 from the `Last.fm` online music system. LFM-1b provides implicit

feedback, user-artist play counts, converted to explicit feedback into the range (1, 5), following the procedure proposed in [144].

Recommender Models

We studied the impact of data properties on the effectiveness of the attacks against the following CF recommendation models:

- **MF** [139] uses the matrix factorization (MF) model as the core predictor that factorizes the user-item preference matrix to learn users' preferences by fitting the previously observed interactions. We set the number of hidden factors (h) to 100, the default value in [127].

- **User-*k*NN** [137] computes the unknown preference score $\hat{\mathbf{s}}_{ui}$ for user u and item i as an aggregate of the ratings of the users who have rated item i and are most similar to user u.

$$\hat{s}_{ui} = b_{ui} + \frac{\sum_{v \in \mathcal{U}_i^k(u)} \text{dist}(u,v) \cdot (s_{vi} - b_{vi})}{\sum_{v \in \mathcal{U}_i^k(u)} \text{dist}(u,v)} \tag{3.12}$$

  where $b_{ui} = \mu + b_u + b_i$, and μ, b_u, b_i respectively are the overall average rating, the observed bias of user u and item i, and $\mathcal{U}_i^k(u)$ is the set of the k closest users to u that have interacted with the same item i.

- **Item-*k*NN** [137] calculates $\hat{\mathbf{s}}_{ui}$ as an aggregate of the ratings of the items, which are most similar to item i.

$$\hat{s}_{ui} = b_{ui} + \frac{\sum_{j \in \mathcal{I}_u^k(i)} \text{dist}(i,j) \cdot (s_{uj} - b_{uj})}{\sum_{j \in \mathcal{I}_u^k(i)} \text{dist}(i,j)} \tag{3.13}$$

  where and $\mathcal{I}_u^k(i)$ denotes the items rated by user u most similar to item i.

For both kNN approaches, we used the formulations that adjust user and item effects — systematic inclinations for some users to provide higher ratings than others, and for some items to collect ratings higher than others items — subtracting biases (i.e., b_{vi}, b_{uj}) from each rating [138]. We set the number of neighbors k equal to 40, and we used the Pearson correlation as the metric to implement the $dist(\cdot)$ function.

Shilling Attack Strategies

We explore the six popular shilling attack strategies to study the impact of data characteristics on the performance of each attack independently: Random attack, Love-hate attack, Bandwagon attack, Popular attack, Average attack, Perfect-knowledge attack. We provide the technical description of each attack in Table 2.2.

Algorithm 1 Sample generation procedure

1: **Input:** $\mathcal{R}$
2: **Results:** $\mathcal{N}$ sub-datasets ($\mathcal{R}_n$)
3: $n \leftarrow 1$
4: **while** $n \leq \mathcal{N}$ **do**
5: Random shuffle the row of the $\mathcal{R}$
6: $num_{users} \leftarrow rnd([100, 2500])$
7: $num_{items} \leftarrow rnd([100, 2500])$
8: $\mathcal{R}_n \leftarrow$ Selection of num_{users}, num_{items} from $\mathcal{R}$
9: **if** $density(urm_n) \in [0.0005, 0.01]$ **then**
10: $n \leftarrow n+1$

Procedure for the Generation of Data Samples

Based on the regression-based explanatory model formalized in Equations (3.10) and (3.11), the goal is to solve regression model coefficients using characteristics generated from various datasets with different structures and content values. The scale and diversity of datasets can significantly impact the accuracy of coefficients computed and, more importantly, on the generalizability of final findings. Toward this aim, motivated by the approach presented in [5], we adopt a sample (i.e., dataset) generation strategy where for a given original dataset, the goal is to generate N different samples(i.e., smaller dataset $\mathcal{R}_n$) with different characteristics. The sampling procedure is specified in Algorithm 1.

For a given recommendation model in User-kNN, Item-kNN, and MF), an attack strategy between the six in Table 2.2, an attack size in $\{1\%, 2.5\%, 5\%\}$, and a dataset in ML-20M, Yelp, and LFM-1b; we generate $\mathcal{N} = 600$ sub-samples resulting in a total number of 162 study cases (i.e., 54 for each attack size) obtained by performing 97,200 attack simulation experiments. We force the densities of the generated $\mathcal{R}_n$ to be in a predefined range of $[0.0005, 0.01]$) to obtain realistic density values. Table 3.2 summarizes the statistics related to each IV (data characteristics) for the 600 generated data-samples.

Reproducibility Details

Before building the regression model, the dataset characteristics are mean-centered. We set the length of the recommendation list to 10 (i.e., $K = 10$) for all experiments. We execute experiments considering three quantities of added fake users equal to $1\%, 2.5\%$, and 5%, of the number of the users in each data sample. However, since a larger attack size is impactful in every condition, it is less meaningful to analyze the impact

Table 3.2 Statistics of Independent Variables averaged across the number of dataset sub-samples ($\mathcal{N} = 600$).

Data	IVs	Min	Max	Mean	σ
ML-20M	$SpaceSize_{log}$	2K	2M	594K	537K
	$Shape_{log}$	0.366	30.039	2.985	2.773
	$Density_{log}$	0.010	0.070	0.019	0.007
	$Gini_{user}$	0.266	0.631	0.547	0.059
	$Gini_{item}$	0.528	0.831	0.737	0.052
	Std_{rating}	0.902	1.183	1.050	0.030
Yelp	$SpaceSize_{log}$	240	3M	618K	695K
	$Shape_{log}$	0.331	3.509	1.225	0.510
	$Density_{log}$	0.002	0.071	0.007	0.007
	$Gini_{user}$	0.052	0.563	0.390	0.089
	$Gini_{item}$	0.068	0.634	0.432	0.090
	Std_{rating}	0.988	1.299	1.151	0.035
LFM-1b	$SpaceSize_{log}$	168	589K	98K	120K
	$Shape_{log}$	0.800	9.685	2.444	1.026
	$Density_{log}$	0.004	0.085	0.016	0.014
	$Gini_{user}$	-0.000	0.422	0.255	0.088
	$Gini_{item}$	0.121	0.819	0.590	0.124
	Std_{rating}	0.577	1.204	0.950	0.077

K = thousand, M = milion

of data characteristics when attacks are consistently effective in all experimental cases. Therefore, we focus our attention only on the smaller size of injected profiles (1%). Finally, we select the number of attacked items as the 0.05% of the number of items in each data sample. To ensure a general analysis of the framework, inspired by [9], we randomly select the same number of target items from all items' popularity quartiles.

Evaluation of EM

While prior research in shilling attacks on RS largely focuses on predictive modeling tasks, in this chapter, we build an explanatory statistical model with the goal to *validate the hypothesis* if there exists an underlying relationship between data characteristics and the explanatory model output $\Delta_{HR@K}(\mathcal{I}_T,\mathcal{U}_T)$. After validating this hypothesis, our secondary goal is to compute the *significance* and *directionality* of this relationship. Thus, the evaluation metrics presented here aim toward assessing the outcome of the explanatory model:

- **Coefficient of determination (R^2)** is a common metric in statistics to measure how well the data fit a (linear) regression model [185]. R^2 represents the proportion of variation in the DV that the IVs can explain. R^2 values range from 0 to 1, 1 means that the DV is completely explained by IVs, while 0 indicates that the model explains none of the variability in the output. For instance, an R^2 of 0.58 means that IVs explains the 58% of variations in the DV.[2]

- **Significance of measured regression coefficients** is measured through the *p-value* for each regression coefficient tests the null hypothesis that the coefficient is equal to 0 (i.e., the IV does not influence the DV). A small *p-value* ($p < 0.05$) indicates that there is enough evidence to reject the null hypothesis (i.e., an effect), and we can assert that the findings are "statistically significant". To help the reader, in Tables 3.3 and 3.4, we use the signs * ($p < 0.05$), ** ($p < 0.01$) and *** ($p < 0.001$) to report which of the coefficient computed are statistically significant. We rely only on statistically significant results in presenting a discussion about the results and drawing the final conclusions.

- **Directionality of the measured regression coefficients** is the sign of the regression coefficient indicates whether there is a positive relation between variation on an IV and DV or a negative relationship. A system designer might use this information to understand and anticipate potential variations in the robustness performance against shilling attacks of the maintained RS.

3.3.2 Results and Discussion

To better understand the merits of the proposed explanatory framework, we aim to answer the following research questions through the course of experiments:

RQ1 Is there an underlying relationship between the presented set of dataset characteristics (IVs) computed on $\mathcal{R}$ and the effectiveness of shilling attack on CF models (DV) measured in terms of $\Delta_{HR@K}(\mathcal{I}_T,\mathcal{U}_T)$?

RQ2 How *significant* is the impact of each IV on the effectiveness of shilling attacks, measured in terms of $\Delta_{HR@K}(\mathcal{I}_T,\mathcal{U}_T)$? What is the *directionality* of this impact (positive or negative)?

[2] In explaining the results presented in Tables 3.3 and 3.4, we rely on ($adj.R^2$) a modified version of R^2, which unlike the latter is not affected by new features added rather if the new feature truly contributes to the overall performance.

RQ3 Do the demonstrated IVs present a consistent behavior when data samples are combined from datasets of all domains (i.e., a domain-independent behavior)?

Within-Dataset Analysis (RQ1-2)

In this section, first, we answer RQ1 to identify if there is an underlying relationship between the described set of IVs computed from $\mathcal{R}$ and the DV, then, we answer RQ2 to study how much the data characteristics can impact variations of $\Delta_{HR@K}(\mathcal{I}_T,\mathcal{U}_T)$ in terms of the significance and directionality. Table 3.3 reports the results to lead this study.

Analysis of Regression Results (RQ1). Given a dataset, a recommendation model, and an attack strategy, we build an explanatory-regression model to explain the relationship between the six IVs and the DV. Regression results for the within-dataset analysis across different dimensions are summarized in Table 3.3. The results obtained for the adjusted coefficient of determination ($adj.R^2$) in Table 3.3 reveal that the six dataset characteristics can explain more than 60% of the variation in $\Delta_{HR@K}(\mathcal{I}_T,\mathcal{U}_T)$ irrespective of the attack type, CF model, and domain (dataset). For instance, by focusing at one randomly selected attack (e.g., the Popular attack), against User-kNN, Item-kNN, and MF models on samples extracted from ML-20M, one can note that the six IVs can respectively explain 85.9%, 91.2%, and 77.2% of the variations in $\Delta_{HR@K}(\mathcal{I}_T,\mathcal{U}_T)$. The corresponding $adj.R^2$ values for three models on Yelp are 78.4%, 75.9%, and 86.4%, and for LFM-1b 66.5%, 65.5% and 78.1%. The $adj.R^2$ coefficient reaches maximum values for the MF model on samples extracted from Yelp ($adj.R^2 > 85\%$), while minimum on User-kNN for LFM-1b ($66\% < adj.R^2 < 67\%$). These results provide (strong) empirical evidence to support the hypothesis that the six identified IVs can explain a substantial portion of the variations in the attack impact measured by $\Delta_{HR@K}(\mathcal{I}_T,\mathcal{U}_T)$ independently of <attack, dataset, model> combination. The explanatory power is highest for MF (when comparing the global behavior of each CF model). However, not a similar observation could be made in favor of a singular attack strategy.

Analysis of Constant Term (RQ2). The constant term represents the *expected attack impact* measured in terms of $\Delta_{HR@K}(\mathcal{I}_T,\mathcal{U}_T)$ for a given <attack, CF-model, dataset> triplet. For example, considering the random attacks on User-kNN, for a random sample (sub-dataset) with average dataset characteristics extracted from ML-20M, Yelp, and LFM-1b, the expected $\Delta_{HR@K}(\mathcal{I}_T,\mathcal{U}_T)$ are 0.179, 0.609 and 0.717,

Table 3.3 Table reporting the regression results for the within dataset analysis (RQ1, RQ2). For a matter of space, we report only the values for the attack size set to 1% of the number of profiles in each sub-sample. We use the following convention to report the statistical significance of the coefficients, i.e., $***p \leq .001$, $**p \leq .01$, $*p \leq .05$.

$\Delta_{HR@10}$		User-*k*NN			Item-*k*NN			MF		
		ML-20M	**Yelp**	**LFM-1b**	**ML-20M**	**Yelp**	**LFM-1b**	**ML-20M**	**Yelp**	**LFM-1b**
Random	$R^2(adj.R^2)$	0.761(0.758)	0.838(0.835)	0.673(0.668)	0.820(0.818)	0.815(0.812)	0.666(0.662)	0.843(0.841)	0.908(0.907)	0.790(0.788)
	$Constant$	**.179*****	**.609*****	**.717*****	.262***	.610***	.715***	.482***	.524***	.688***
	$SpaceSize_{log}$	-0.063***	.041	-0.629***	.008	.003	-0.520***	.040*	.368***	-0.368***
	$Shape_{log}$	.184***	.248***	.288*	.139***	.198***	.125	.207***	.275***	.192
	$Density_{log}$	-0.189***	-0.316*	-1.546***	-0.174***	-0.376**	-1.366***	-0.274***	.393***	-1.047***
	$Gini_{users}$	.277	-0.012	1.901***	-0.223	.030	.891	.178	-0.660**	.988*
	$Gini_{item}$	-0.102	-0.485	1.753***	-0.305	-0.241	1.784***	.102	-1.270***	1.355***
	Std_{rating}	-0.072	.287	-0.152	-0.120	.326	.012	-0.240	.311*	-0.108
Love-Hate	$R^2(adj.R^2)$	0.806(0.803)	0.839(0.837)	0.673(0.668)	0.841(0.839)	0.822(0.820)	0.665(0.661)	0.825(0.823)	0.911(0.910)	0.789(0.787)
	$Constant$	.267***	.657***	.717***	.419***	.662***	.716***	.655***	.578***	.688***
	$SpaceSize_{log}$	-0.027*	.042	-0.628***	.125***	.028	-0.506***	.073***	.393***	-0.364***
	$Shape_{log}$	.209***	.131*	.287*	.103***	.065	.105	.059***	.105*	.194
	$Density_{log}$	-0.198***	-0.290*	-1.544***	-0.071*	-0.316*	-1.337***	-0.209***	.434***	-1.044***
	$Gini_{users}$	.347	.114	1.896***	-0.852***	-0.002	.831	-0.231	-0.920***	.972*
	$Gini_{item}$	-0.430	-0.150	1.754***	-0.583**	.043	1.806***	.985***	-0.764*	1.309***
	Std_{rating}	-0.179	.239	-0.151	-0.188	.259	.022	-0.168	.278	-0.073
Bandwagon	$R^2(adj.R^2)$	0.777(0.774)	0.835(0.833)	0.673(0.668)	0.818(0.816)	0.813(0.810)	0.665(0.661)	0.841(0.839)	0.914(0.912)	0.786(0.784)
	$Constant$	.180***	.607***	.717***	.244***	.609***	.715***	.435***	.522***	.689***
	$SpaceSize_{log}$	-0.068***	.040	-0.635***	-0.015	-0.002	-0.514***	-0.006	.372***	-0.366***
	$Shape_{log}$	.190***	.266***	.293*	.145***	.212***	.116	.244***	.278***	.206*
	$Density_{log}$	-0.188***	-0.305*	-1.559***	-0.192***	-0.382**	-1.354***	-0.314***	.401***	-1.047***
	$Gini_{users}$	.342	.110	1.919***	-0.080	.104	.869	.602***	-0.680**	.976*
	$Gini_{item}$	-0.041	-0.483	1.755***	-0.158	-0.251	1.797***	.268	-1.278***	1.276***
	Std_{rating}	-0.036	.284	-0.151	-0.087	.315	.016	-0.290	.321*	-0.066
Popular	$R^2(adj.R^2)$	0.860(**0.859**)	0.787(**0.784**)	0.670(**0.665**)	0.913(**0.912**)	0.762(**0.759**)	0.660(**0.655**)	0.774(**0.772**)	0.866(**0.864**)	0.784(**0.781**)
	$Constant$	.589***	.810***	.724***	.537***	.788***	.705***	.724***	.775***	.694***
	$SpaceSize_{log}$	-0.020	-0.411***	-0.617***	.099***	-0.441***	-0.517***	.009	-0.238**	-0.391***
	$Shape_{log}$	.187***	.028	.268*	.176***	.084	.118	.084***	.041	.210*
	$Density_{log}$	-0.335***	-1.175***	-1.521***	-0.162***	-1.261***	-1.361***	-0.337***	-0.898***	-1.092***
	$Gini_{users}$	.225	1.050**	1.872***	-0.318	1.129**	.879	-0.055	.199	1.082*
	$Gini_{item}$	.491	1.735***	1.764***	-0.225	1.305***	1.715***	1.346***	1.195**	1.347***
	Std_{rating}	-0.182	.002	-0.129	-0.353*	.049	.017	-0.043	.237	-0.062
Average	$R^2(adj.R^2)$	0.759(0.756)	0.831(0.829)	0.673(0.668)	0.819(0.816)	0.813(0.811)	0.666(0.661)	0.845(0.843)	0.910(0.909)	0.790(0.788)
	$Constant$	.187***	.609***	.717***	.276***	.608***	.715***	.502***	.523***	.690***
	$SpaceSize_{log}$	-0.063***	.048	-0.632***	.018	.010	-0.513***	.046**	.373***	-0.339***
	$Shape_{log}$	.182***	.260***	.291*	.136***	.201***	.114	.189***	.273***	.167
	$Density_{log}$	-0.189***	-0.290*	-1.553***	-0.162***	-0.359**	-1.352***	-0.271***	.405***	-0.991***
	$Gini_{users}$	.296	.074	1.907***	-0.265	.028	.857	.095	-0.652**	.833*
	$Gini_{item}$	-0.072	-0.522	1.755***	-0.284	-0.243	1.796***	.258	-1.267***	1.317***
	Std_{rating}	-0.065	.299	-0.150	-0.114	.312	.019	-0.242	.322*	-0.079
Perfect Knowledge	$R^2(adj.R^2)$	0.790(0.787)	0.836(0.834)	0.676(0.671)	0.828(0.826)	0.823(0.821)	0.670(0.666)	0.847(0.845)	0.914(0.913)	0.793(0.790)
	$Constant$	.267***	.608***	.717***	.275***	.603***	.711***	.479***	.519***	.688***
	$SpaceSize_{log}$	-0.039**	.071	-0.609***	.006	.027	-0.504***	.034*	.382***	-0.358***
	$Shape_{log}$	.187***	.256***	.271*	.137***	.247***	.104	.207***	.275***	.198*
	$Density_{log}$	-0.139***	-0.247	-1.508***	-0.181***	-0.335**	-1.337***	-0.277***	.427***	-1.032***
	$Gini_{users}$	.399	-0.067	1.815***	-0.184	.093	.789	.240	-0.714**	.969*
	$Gini_{item}$	.270	-0.582	1.746***	-0.226	-0.506	1.771***	.216	-1.284***	1.276***
	Std_{rating}	-0.135	.269	-0.132	-0.142	.342	-0.004	-0.289	.290	-0.085

respectively. The knowledge about expected performance can give the system designer predictive knowledge about attacks' impacts under average conditions. However, it remains outside the main focus of this work, as we aim for explanatory performance (rather predictive) of the system; we nevertheless report these results for the sake of completeness.

Analysis of Impact of data characteristics (RQ2). The first observation is that unlike the findings in [5], which show a consistent significant behavior for all the IVs mentioned above in the explanation of the general performance of RS (not for shilling attacks), the significance of the computed regression coefficients for the IVs tends to vary for each IV or group of IVs. The results show that the regression coefficients computed for the structural $\mathcal{R}$ characteristics (i.e., $SpaceSize_{log}$, $Shape_{log}$, $Density_{log}$) are statistically significant. This suggests that there is enough statistical evidence to support the hypothesis that structural $\mathcal{R}$ characteristics can explain the variations in the DV ($p < \{0.05, 0.01, 0.001\}$), which is equal to state that there is an underlying relationship between these three IVs and the DV. However, results for the other IVs vary depending on <attack, CF-model, dataset> triplet, or insignificant as in the case of Std_{rating}. For instance, the coefficients for Gini indices (i.e., $Gini_{user}$ and $Gini_{item}$) are most significant for shilling attacks against MF, particularly for samples drawn from the Yelp and LFM-1b datasets. The coefficients for Std_{rating} are insignificant ($p\text{-}value > 0.05$) in all experimented cases, except for two cases <Random/Average attack, MF, Yelp>, implying that this dataset characteristic, which deals directly with rating values of the $\mathcal{R}$, plays an insignificant role on the impact of shilling attacks against CF models.

In summary, the results of the within-dataset analysis provide strong statistical evidence that structural $\mathcal{R}$ characteristics (i.e., $Shape_{log}$, $SpaceSize_{log}$, $Density_{log}$) play a pivotal role in the impact of attacks targeted on CF models for all cases in the <attack, CF, dataset> triplet; rating frequency features play a significant role mostly for attacks targeted on model-based MF recommendation. Finally, the role of Std_{rating} features that deal directly with rating values cannot be confirmed, since they have shown no evidence of having a significant impact.

Given the statistical significance of the regression coefficients for many IVs, the next step is to explore the *directionality* of this impact. Results summarized in Table 3.3 show that the effect of $Density_{log}$ is **negative** on $\Delta_{HR@K}(\mathcal{I}_T, \mathcal{U}_T)$ across majority of the cases in <attacks, CF-model, dataset> triplet (except the ones on <MF, Yelp>. This result is interesting and is consistent with findings in RS literature that increasing

the density (or decreasing sparsity) of the $\mathcal{R}$ not only improves the general performance of CF models (as recognized in the prior research [69, 5]), but also **reduces** the likelihood of attacks' effectiveness. One plausible explanation can be as follows: if we fix [3] the number of users and items and increase the number of genuine ratings (e.g., asking to rate more), the accuracy of similarities computed is improved due to using more genuine ratings. As these similarities are generally vulnerable to the insertion of fake profiles, adding more genuine feedbacks can help to decrease the impact of attacks.

Additionally, we can note that $SpaceSize_{log}$ has a **negative** impact on $\Delta_{HR@K}(\mathcal{I}_T,\mathcal{U}_T)$ in **neighborhood models**, which means that increasing the space size of $\mathcal{R}$ makes neighborhood models less vulnerable against attacks. Furthermore, higher $SpaceSize_{log}$ (under fixed sparsity) means more users, items, and ratings. This provides neighborhood models with more non-malicious candidate users (and items) to compute similarities, and can reduce the effect of attacks. Finally, and on the contrary, $Shape_{log}$ presents a consistently positive influence on the efficacy of the attacks. This is a novel insight. We can explain it by considering the following example: increasing $Shape_{log}$ leads to an increased number of users with respect to items (i.e., decreasing items). In this way, it could be easier to push the target item to higher positions inside the recommendation list (i.e., fewer items contribute to the recommendation).

Between-Dataset Analysis (RQ3)

The goal of the within-dataset analysis presented in the previous section was to investigate the impact of data characteristics on shilling attacks for each <attack, CF-model, dataset> triplet. In this section, we aim to provide a *domain-independent* analysis of the same study (impact of data characteristics on attacks' effectiveness) by combing rating scores of all three datasets. The regression model and the OLS follow Eq. 3.9 and 3.11, and we replicate the exact procedure described in [5]. Note that the DV here contains rating samples from all three datasets. Results of the between-dataset analysis are summarized in Table 3.4. Here, the $adj.R^2$ values are consistent with those in within-dataset analysis in most experimental cases. For instance, $adj.R^2$ tells us that the selected IVs explain more than 80% of the variation in $\Delta_{HR@K}(\mathcal{I}_T,\mathcal{U}_T)$ independently form <attack, CF-model> pair. Furthermore, results still support that structural $\mathcal{R}$ properties have a statistically significant impact on each CF model. The *p-values* of $SpaceSize_{log}$, $Shape_{log}$, and $Density_{log}$ regression coefficients are less than 0.001 in each pair of experiments. Moreover, the *directionality* analysis of structural IVs in Table 3.4 is consistent with the insights drawn from previous

[3] Note that in providing these examples, we fix other IVs and focus on the impact of an IV.

Table 3.4 Table reporting the regression results for the between dataset analysis (RQ3). For a matter of space, we report only the values for the attack size set to 1% of the number of profiles in each sub-sample. We use the following convention to report the statistical significance of the coefficients, i.e., $^{***}p \leq .001$, $^{**}p \leq .01$, $^{*}p \leq .05$.

$\Delta_{HR@10}$		**User-*k*NN**	**Item-*k*NN**	**SVD**
Random	$R^2(adj.R^2)$	0.832(0.831)	0.814(0.813)	0.843(0.843)
	ML-20M (Constant)	.179***	.262***	.482***
	Yelp	.429***	.347***	.041***
	LFM-1b	.537***	.452***	.204***
	$SpaceSize_{log}$	-0.197***	-0.096***	.047***
	$Shape_{log}$	.153***	.108***	.204***
	$Density_{log}$	-0.729***	-0.550***	-0.253***
	$Gini_{user}$	.552***	-0.008	.101
	$Gini_{item}$	.728***	.439***	-0.032
	Std_{rating}	-0.057	.058	-0.029
Love-Hate	$R^2(adj.R^2)$	0.817(0.816)	0.774(0.773)	0.833(0.832)
	ML-20M (Constant)	.267***	.419***	.655***
	Yelp	.390***	.243***	-0.077***
	LFM-1b	.449***	.295***	.031***
	$SpaceSize_{log}$	-0.142***	.040***	.113***
	$Shape_{log}$	.174***	.090***	.083***
	$Density_{log}$	-0.620***	-0.289***	-0.137***
	$Gini_{user}$	.679***	-0.218	-0.285**
	$Gini_{item}$	.429***	.021	.122
	Std_{rating}	-0.073	.055	-0.032
Bandwagon	$R^2(adj.R^2)$	0.831(0.831)	0.818(0.817)	0.848(0.848)
	ML-20M (Constant)	.179***	.244***	.435***
	Yelp	.427***	.364***	.087***
	LFM-1b	.537***	.470***	.253***
	$SpaceSize_{log}$	-0.199***	-0.118***	.008
	$Shape_{log}$	.161***	.115***	.235***
	$Density_{log}$	-0.730***	-0.591***	-0.331***
	$Gini_{user}$	.589***	.082	.267*
	$Gini_{item}$	.720***	.497***	-0.019
	Std_{rating}	-0.059	.058	-0.018
Popular	$R^2(adj.R^2)$	0.744(0.742)	0.741(0.740)	0.800(0.799)
	ML-20M (Constant)	.589***	.537***	.725***
	Yelp	.222***	.252***	.051***
	LFM-1b	.133***	.166***	-0.032***
	$SpaceSize_{log}$	-0.059***	.051***	.040**
	$Shape_{log}$	.191***	.169***	.111***
	$Density_{log}$	-0.445***	-0.252***	-0.283***
	$Gini_{user}$	.544***	-0.050	-0.140
	$Gini_{item}$	.229*	-0.258*	.288**
	Std_{rating}	-0.124	-0.011	-0.017
Average	$R^2(adj.R^2)$	0.828(0.827)	0.810(0.809)	0.844(0.843)
	ML-20M (Constant)	.187***	.275***	.502***
	Yelp	.421***	.332***	.020***
	LFM-1b	.529***	.438***	.186***
	$SpaceSize_{log}$	-0.193***	-0.082***	.065***
	$Shape_{log}$	.152***	.107***	.192***
	$Density_{log}$	-0.718***	-0.522***	-0.219***
	$Gini_{user}$	.559***	-0.039	.011
	$Gini_{item}$	.717***	.407***	-0.062
	Std_{rating}	-0.054	.059	-0.013
Perfect Knowledge	$R^2(adj.R^2)$	0.812(0.811)	0.813(0.812)	0.847(0.846)
	ML-20M (Constant)	.266***	.274***	.479***
	Yelp	.341***	.328***	.039***
	LFM-1b	.449***	.434***	.207***
	$SpaceSize_{log}$	-0.141***	-0.088***	.049***
	$Shape_{log}$	.167***	.109***	.206***
	$Density_{log}$	-0.613***	-0.540***	-0.250***
	$Gini_{user}$	.479***	-0.035	.087
	$Gini_{item}$	.546***	.387***	-0.048
	Std_{rating}	-0.061	.048	-0.031

study. In summary, for all CF models, $Shape_{log}$ has a positive impact, $Density_{log}$ has a negative influence, while the impact of $SpaceSize_{log}$ is (for most cases) negative on neighborhood recommenders and positive on model-based models. Additionally, these results show that rating frequency values of IVs have not shared a statistically significant impact on the DV.

In summary, the results of the between-dataset analysis support those presented in the within-dataset analysis. Given the heterogeneity of domains and variety of attack and CF models tested, this can be interpreted by the fact that ***effects*** *of data characteristics studied in this chapter are* ***NOT domain-specific*** *and the insights/conclusions obtained from this study can be applied to a broad range of domains for most popular attack and CF models.*

3.4 Related Work

Hand-engineered poisoning of against recommendation models can be categorized based on various dimensions: the intent behind the attack (push or nuke) [163, 143], and the attacker's knowledge, i.e., informed [162, 105] and semantic-enhanced [21, 16] attacks. Numerous research articles has been produced in the context of shilling attacks, which can be broadly categorized into three research directions: (i) attack types [143, 161, 105], (ii) detection strategies [63, 163, 8] and (iii) robustness evaluation [173]. A common characteristic of the prior literature is that they mostly focus on algorithmic and procedural exploration and analysis of attack and defense strategies. The user-rating matrix ($\mathcal{R}$) (and properties extracted from it) is the key information source of CF and attack models. A substantial amount of works explored the effects of different data characteristics measured from $\mathcal{R}$ on recommendation accuracy. For instance, the *sparsity* of the dataset has been widely studied since it largely influences recommendation accuracy [69, 5], and so the *skewness* of data (i.e., the distribution of feedback across items) has been demonstrated to influence the problem of predicting customer behavior and suggesting matching products [32, 123]. However, we have conducted this research believing that there exists a lack of systematic and large-scale analysis of the impact of dataset characteristics (e.g., sparsity, size, rating skewness) on the robustness of collaborative models against shilling attacks. The goal of this chapter has been to fill this gap by investigating the effects of $\mathcal{R}$ data characteristics on an attack performance metric with explanatory-based regression models.

3.5 Summary

In this chapter, we have proposed a model to study the impact of data characteristics on the effectiveness of the most famous shilling attacks against popular CF methods. We have considered a suite of data characteristics, which can be classified according to (i) the structure of the $\mathcal{R}$, (ii) the rating frequency distribution, and (iii) the rating values. We have used a regression-based explanatory model and have relied on statistical significance with informed *p-value* in order to verify the impact of data characteristics. Results of extensive experiments have provided sufficient statistical evidence to accept the hypothesis that, first, the identified data characteristics can account for a considerable portion of variations in attack performance (global perspective) and, second, that there remain considerable differences in the significance (and directionality) of this impact among features. For instance, while $\mathcal{R}$ structural properties (size, shape, density) consistently indicate having an impact on the model output, the rating property (standard deviation of ratings) does not show an effect. On the other hand, distribution properties (Gini user and item) show a higher impact on memory-based models. As the proposed explanatory framework can support a system designer in evaluating the robustness performance by looking at the dataset characteristics, we plan to extend the studied characteristics (e.g., user-item relations), CF models (e.g., deep learning approaches), adversarial machine-learned attacks.

Chapter 4

Semantics-Aware Shilling Attacks

Can public available semantic information be exploited to develop more effective shilling attacks against CF models, where the effectiveness is measured in terms of a raise of the recommendability of the target items in the recommendation lists?

Several fields have benefited from the adoption of knowledge graphs (KGs). In RSs, they have resulted in accurate, personalized recommendations of items in CF models. While the research community has extensively studied KGs to solve various recommendation problems, sufficient attention was not paid to the possibility of exploiting them to compromise the quality of recommendations. KGs provide a rich source of information for item representation and recommendation that can dramatically increase the attackers' knowledge about the victim recommendation platform. To this end, this chapter introduces a new attack strategy, named semantics-aware shilling attack (`SAShA`), that leverages semantic features extracted from a KG. `SAShA` provides the semantics-aware variant of three state-of-the-art attack strategies: Random, Average, and Bandwagon. These improved attacks can exploit graph relatedness measures, i.e., *Katz* and *Exclusivity*-based, computed considering 1-hop and 2-hops of graph exploration. We perform an extensive experimental evaluation with four state-of-the-art recommendation systems and two well-known recommendation datasets to investigate the effectiveness of `SAShA`. Since the semantics of relations has a crucial role in KGs, we also analyze the impact of relations' semantics by grouping them in various classes. Our results indicate the benefit of embracing KGs in favor of the attackers' capability in attacking recommendation systems.

4.1 Introduction

The advent of Knowledge Graphs (KGs) has definitely changed the way structured information is stored. Developed to make the Semantic Web a concrete idea, it has become much more than that. The core idea of building a semantic network in which information is represented as directed labeled graphs (RDF graphs) is disarmingly simple. Nevertheless, thanks to the possibilities it paves, it has been welcomed with several promises and expectancies. Complete interoperability, the ability to link knowledge across domains, and the possibility of exploiting Logical inference and proofs are just a few of them. In numerous domains, the exploitation of the Knowledge Graph information has become the norm. Thanks to the appearance of wide-ranging Linked Datasets like `DBpedia` and Wikidata, we have witnessed the flourishing of novel techniques in several research fields, like Machine Learning, Information Retrieval, and Recommender Systems.

Interestingly, despite the astonishing spread of KGs, little attention has been paid to knowledge-aware strategies to mine RS's security. In a Web always composed of unstructured information, KGs are the pillars of the Semantic Web. They have become increasingly important to represent data employing a flexible and interoperable semantic graph data structure. Several well-known tools have been built on KGs, like IBM Watson [41], public decision-making systems [196], and advanced machine learning techniques [66, 22]. Additionally, the Linked Open Data (LOD) initiative[1] has given birth to a broad ecosystem of linked data datasets known as LOD-cloud[2]. These KGs provide comprehensive information on numerous knowledge domains. Consequently, if a malicious agent decides to attack one of these domains, items' semantic descriptions would be inestimable.

In the chapter, we investigate the possibility of improving an attack's efficacy by leveraging semantic knowledge. One significant contribution of the chapter is exploiting publicly available information obtained from KG to generate more influential fake profiles to threaten CF models' performance. The resulting attack strategy is named semantics-aware shilling attack `SAShA`. Beyond the definition of `SAShA` strategy, our contribution is to extend state-of-the-art shilling attack approaches such as Random, Bandwagon, and Average profiting from semantic knowledge shown in Table 4.1. Remarkably, the attacks' semantics-enhanced variants only rely on publicly available information without supposing any additional knowledge about the system.

[1] https://data.europa.eu/euodp/en/linked-data
[2] https://lod-cloud.net/

The core idea is to reformulate the attacks with the rationale of considering the semantic similarity between the target item with the other items in the catalog. The intuition of the approach is that semantic similarity (or, more broadly, semantic relatedness) can safely suffice the lack of the system's knowledge to craft natural and coherent fake profiles. These profiles are indistinguishable from the real ones, and they effortlessly enter the neighborhood of users and items.

We investigate `SAShA` using the famous (but semantics-unaware) cosine similarity, the *Katz centrality*, and *Exclusivity-based relatedness* between the semantic description of items. Then, we explore KGs until the second hop, providing a much more in-depth investigation of semantic descriptions' role for this task. Finally, to provide a more fine-grained analysis, we have grouped the semantic relations into three classes: ontological, categorical, and factual relations.

In detail, this chapter proposes novel methods for the integration of semantics in the shilling attacks addressing the following research directions:

- three novel graph topological and semantic approaches to build the set of products from which the adversary can craft the fake profiles;
- an extensive study of the efficacy of the attack considering a two-hops graph exploration, and involving a state-of-the-art deep neural recommendation model;
- novel semantic shilling attack strategies based on Random, Average, and Bandwagon standard strategies;
- a deeper discussion of the experimental results involving several dimensions: number of explored hops, type of considered relation, recommendation model, amount of injected fake profiles, and dataset;
- the publication of the full experimental framework and the pre-processed datasets that can be used, out-of-the-box, for further investigations.

Experiments described in this chapter evaluate the impact of proposed attacks against the recommendation models. To this end, we have exploited two real-world recommender systems datasets (`LibraryThing` and `Yahoo!Movies`). Experimental results sharply indicate that KG information is a valuable source of knowledge that improves attacks' effectiveness. Moreover, the adoption of semantic relatedness measures can unleash the full potential of the semantics-aware attacks.

The remainder of the chapter proceeds as follows. Section 4.2 describes the proposed approach (`SAShA`), introduces the semantic relatedness measures, and formalizes the

semantic attack strategies. Section 4.3 focuses on the experimental validation of the proposed attack scenarios. We also provide an in-depth discussion of the experimental results, analyzing the several dimensions of the study. Then, in Section 4.4, we provide an overview of the state-of-the-art of application of KGs in RS. Finally, in Section 4.5, we draw some conclusions and introduce the open challenges.

4.2 Method

This section introduces the reader to the notations and formalism that may help understand the design of shilling attacks against targeted items integrating information obtained from a knowledge graph (KG). First, we focus on categorizing the predicates in a KG and formalizing the semantic features' extraction considering a single- and double-hop exploration of the KG (Section 4.2.1). Then, the adopted relatedness measures are summarized (Section 4.2.2). Finally, semantics-aware extensions to various widespread shilling attacks, namely: Random, Average, and Bandwagon attacks in Section 4.2.3.

4.2.1 Knowledge Graph Content Extraction

A KG is a structured repository of knowledge, designed in the form of a graph, that encodes various kinds of information:

- **Factual.** General statements as *Rika Dialing was born in Crete* or *Heraklion is Crete's capital* that describe an entity by using a controlled vocabulary of predicates that connect the entity to other entities (or literal values);
- **Categorical.** These statements connect the entity to a particular category (i.e., the categories associated with a Wikipedia page). Often, categories are in turn organized as a hierarchy;
- **Ontological.** These are formal statements that describe the entity's nature and its ontological membership to a specific class. Classes are often organized in a hierarchical structure. In contrast to categories, sub-classes and super-classes are connected through `IS-A` relations.

In a knowledge graph, we can express statements through triplets $\sigma \xrightarrow{\rho} \omega$, with a *subject* ($\sigma$), a *predicate (or relation)* (ρ), and an *object* (ω). There are several ways to transform the knowledge coming from a knowledge graph into a feature. We have chosen to represent each distinct path as an explicit feature [24]. In the next section, it will be

clear why it is convenient. Given a set of items $\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_N\}$ in a collection and the corresponding triples $\langle i, \rho, \omega \rangle$ in a knowledge graph, the set of 1-hop features is defined as $1\text{-}HOP\text{-}F = \{\langle \rho, \omega \rangle \mid \langle i, \rho, \omega \rangle \in \mathcal{KG} \text{ with } i \in I\}$.

In an analogous way, we can identify 2nd-hop features. By continuing the exploration of KG we retrieve the triples $\omega \xrightarrow{\rho'} \omega'$, where ω is the *object* of a 1st-hop triple and the *subject* of the next triple. The double-hop *predicate* is denoted by ρ' and the *object* is referred to as (ω'). Therefore, the overall feature set is defined as $2\text{-}HOP\text{-}F = \{\langle \rho, \omega, \rho', \omega' \rangle \mid \langle i, \rho, \omega, \rho', \omega' \rangle \in \mathcal{KG} \text{ with } i \in I\}$. Given the current definition, 2nd-hop features also contain heterogeneous predicates (see the previous classification of different kinds of statements). To make it possible to analyze the impact of the kind of semantic information, we consider a 2nd-hop feature as Factual *if and only if* both relations (ρ, and ρ') are Factual. The same holds for the other types of encoded information.

4.2.2 Entity Similarity/Relatedness in KGs

The keystone of the KG representation is the semantics enclosed in the resource description and the predicates that connect the different resources. Nevertheless, if the metric to compute similarities between the resources is not carefully chosen, this piece of information is lost irretrievably. Motivated by this awareness, we decided to consider a broad spectrum of diverse similarity/relatedness metrics: **Cosine Vector Similarity** [86], **Katz's centrality** [135], and **Exclusivity-based** [128] semantic relatedness. The three metrics cover three different aspects of the similarity between the resources: a signal of the overlap of the descriptions, the average length of the paths that connect the resources, and a semantics-aware signal that highlights the relations between the resources.

Cosine Vector Similarity

is a well-known similarity that is very popular in recommendation systems. The idea is to measure how similar the two different representations are. Suppose a numerical vector can represent the resource description, with the number of the predicate-object chains observed in KG being the vector's cardinality. Mathematically, it measures the cosine of the angle between two vectors that represent two different resources. The smaller the angle, the higher is the cosine, and thus the similarity. Suppose i and j are two items in the KG, and $F(\cdot)$ is a function that returns the features associated with an entity in the KG. Hence, $in(i, f)$ is a function that returns 1 if entity i is associated

with feature f, else 0. The Cosine Vector Similarity has been already formulated for KG as follows [86]:

Katz's centrality

[135] is a famous graph-centrality measure that inspired several semantics-aware metrics [171, 128]. Katz suggests that the probability of the path between two nodes can indicate the effectiveness of the link. Given a constant probability for a single-hop path, called α, the whole path's overall probability is α^y, where y is the number of the nodes involved. Hulpus [128] exploits the rationale to build a relatedness measure. Therefore, he defined the Katz relatedness between two items i and j as the accumulated score over the top-t-shortest paths between them.

$$rel^{(t)}_{Katz}(i,j) = \frac{\sum_{p \in SP^{(t)}_{ij}} \alpha^{length(p)}}{t} \tag{4.1}$$

where $SP^{(t)}_{ij}$ is the set of the top-t-shortest paths between items i and j.

$$sim(i,j) = \frac{\sum_{f \in F(i) \cup F(j)} in(i,f) \cdot in(j,f)}{\sqrt{\sum_{f \in F(i)} in(i,f)^2} \cdot \sqrt{\sum_{f \in F(j)} in(j,f)^2}} \tag{4.2}$$

Exclusivity-based semantic relatedness

[128] is a semantic relatedness measure that takes into account the type of relations that connect two nodes. The idea is that two concepts are strongly connected if the type of relations between them is different from the type of relations they have with other concepts. This property of relations, named Exclusivity, is defined as follows.

Suppose a predicate ρ of type τ between two items i and j, directed from i to j. The Exclusivity of predicate ρ is the probability to select, with a uniform random distribution, a predicate ρ' of type τ among the predicates of type τ that exit resource i and enter node j, such that predicate ρ' is exactly the predicate ρ:

$$exclusivity(i \xrightarrow{\tau} j) = \frac{1}{|i \xrightarrow{\tau} *| + |* \xrightarrow{\tau} j| - 1} \tag{4.3}$$

where $|i \xrightarrow{\tau} *|$ denotes the cardinality of relations of type $\tau \in \mathcal{T}$ that exit resource i, and $|* \xrightarrow{\tau} j|$ denotes the number of relations of type $\tau \in \mathcal{T}$ that enter resource j. Since the relation $i \xrightarrow{\tau} j$ is in $|i \xrightarrow{\tau} *|$ and in $|* \xrightarrow{\tau} j|$, 1 is subtracted from the denominator.

Table 4.1 Overview of `SAShA` shilling attack strategies and their profile composition for adversaries' goal of *pushing* a target item ($\mathcal{I}_T$).

	Selected Items ($\mathcal{I}_S$)		**Filler Items** ($\mathcal{I}_F$)			$\mathcal{I}_\phi$	$\mathcal{I}_T$
Attack Type	**Number Items**	**Rating**	**Selection**	**Number Items**	**Rating**		
`SAShA` Random	$\emptyset$		Semantics-aware	$\frac{\sum_{u\in\mathcal{U}}\lvert\mathcal{I}_u\rvert}{\lvert\mathcal{U}\rvert}-1$	$rnd(N(\mu,\sigma^2))$	$\mathcal{I}-\mathcal{I}_F$	max
`SAShA` Love-Hate	$\emptyset$		Semantics-aware	$\frac{\sum_{u\in\mathcal{U}}\lvert\mathcal{I}_u\rvert}{\lvert\mathcal{U}\rvert}-1$	min	$\mathcal{I}-\mathcal{I}_F$	max
`SAShA` Average	$\emptyset$		Semantics-aware	$\frac{\sum_{u\in\mathcal{U}}\lvert\mathcal{I}_u\rvert}{\lvert\mathcal{U}\rvert}-1$	$rnd(N(\mu_f,\sigma_f^2))$	$\mathcal{I}-\mathcal{I}_F$	max
`SAShA` Bandwagon	$(\frac{\sum_{u\in\mathcal{U}}\lvert\mathcal{I}_u\rvert}{\lvert\mathcal{U}\rvert})/2-1$	max	Semantics-aware	$(\frac{\sum_{u\in\mathcal{U}}\lvert\mathcal{I}_u\rvert}{\lvert\mathcal{U}\rvert})/2$	$rnd(N(\mu,\sigma^2))$	$\mathcal{I}-\mathcal{I}_S-\mathcal{I}_F$	max

where (μ, σ) are the dataset average rating and rating variance, (μ_f, σ_f) are the filler item $\mathcal{I}_F$ rating average and variance, and min and max are the minimum and maximum rating value. rnd function generates one integer (i.e., rating) from a discrete uniform distribution.

The exclusivity score for a predicate falls inside the $(0,1]$ interval. The value 1 denotes the extreme case in which the predicate is the only relation of its type for both i and j.

Given a path through KG, $\mathcal{P} = n_1 \xrightarrow{\tau} n_2 \xrightarrow{\tau_2}, \ldots, n_k$ with $\tau_i \in \mathcal{T}^{\mp}$, the weight of the path is defined as:

$$weight(\mathcal{P}) = \frac{1}{\sum_i \frac{1}{exclusivity(n_i \xrightarrow{\tau_i} n_{i+1})}} \tag{4.4}$$

Finally, the relatedness between two resources can be computed as the sum of the path weights of the top-t paths between the resources with the highest weights. To penalize longer paths, a constant length decay factor, $\alpha \in (0,1]$, can be introduced. The overall exclusivity-based relatedness measure is therefore defined as follows:

$$rel^{(t)}_{Excl}(i,j) = \sum_{\mathcal{P}_n \in P^t_{ij}} \alpha^{lenght(\mathcal{P}_n)} weight(\mathcal{P}_n) \tag{4.5}$$

4.2.3 `SAShA` Strategies

Previous works on shilling attacks against RS models have predominately focused on CF models, and the way the user interaction data (ratings) can be exploited to craft more effective shilling profiles (see Table 2.2). In our view, a rich source of knowledge, namely KGs, has been neglected in the design of such attacks. To fill this gap, in this chapter, we strengthen state-of-the-art attack strategies by exploiting semantic similarities between items. The main idea behind our proposed semantics-aware shilling attack (`SAShA`) strategies is that we can compute the similarity/relatedness between the target $\mathcal{I}_T$ with other items in the catalog by exploiting the features extracted from a KG. This semantic information is used to construct the filler set $\mathcal{I}_F$, by semantically selecting the items. The key insight in the proposed approach is that the exploitation of semantic similarities/relatedness leads to the generation of more natural and coherent

fake profiles, given that the representative description of items is encoded in computing pairwise item similarities. Table 4.1 present the semantic extension of the classical hand-engineered shilling attacks presented in Chapter 2. Further details are presented below.

- **Semantics-aware Random Attack** is an extension of the baseline Random Attack [143]. The baseline version is naive attack, which uses randomly chosen items ($\alpha = 0, \phi =$ *profile-size*) to create a fake user profile. The ratings attributed to $\mathcal{I}_\phi$ are sampled from a uniform distribution (see Table 4.1). We modify this attack by selecting the items to complete $\mathcal{I}_F$ with the proposal semantics-aware technique. For this purpose, we compute semantic similarities/relatedness between the items in the catalog e the target item using KG-based features (cf. Section 4.2.1). Afterward, we identify the most similar items ($\mathcal{I}_T$) by considering the first quartile of most similar items, and we extract ϕ items from this set by adopting a uniform distribution.

- **Semantics-aware Average Attack** is an informed attack strategy that extends the AverageBots attack [163]. The baseline attack leverages the mean and variance of the ratings, which is then used to sample each filer item's rating from a normal distribution built using these values. Similar to the previous semantics-aware attack extension, we extract the filler items by exploiting semantic similarities derived from a KG. Finally, as before, we consider the items in the first quartile of the most semantically similar/related to $\mathcal{I}_T$ as the candidate filler items ($\mathcal{I}_F$).

- **Semantics-aware Bandwagon Attack** is a low-knowledge attack that extends the standard Bandwagon attack [174]. We leave unchanged the injection of the selected items ($\mathcal{I}_S$), which are the most popular ones and on which we associate the maximum possible rating (see Table 4.1). However, similarly to the previous two semantic attack extensions, we complete $\mathcal{I}_F$ by taking into account the semantic similarity/relatedness between the target item $\mathcal{I}_T$ and the rest of the catalog.

- **Semantics-aware Love-Hate Attack** is a low-knowledge attack that extends the standard Love-Hate attack [163]. This attack randomly extracts filler items $\mathcal{I}_F$ from the catalog. All these items are associated with the minimum possible rating value. The Love-Hate attack aims to reduce the average rating of all the platform items but the target item. Indeed, even though the target item is not present in the fake profiles, its relative rank increases. We have re-interpreted

Table 4.2 Datasets statistics.

Dataset	#Users	#Items	#Ratings	Sparsity	#F-1Hop	#F-2Hops
`LibraryThing`	4,816	2,256	76,421	99.30%	56,019	4,259,728
`Yahoo!Movies`	4,000	2,526	64,079	99.37%	105,733	6,697,986

the rationale behind the Love-Hate attack by taking into account the semantic description of the target item and its similarity with other items within the catalog. In this case, we extract items to fill $\mathcal{I}_F$ from the 2nd, 3rd, and 4th quartiles. As in the original variant, the rationale is to select the most dissimilar items. Note that in this chapter, we do not investigate the semantics-aware extension of the Love-Hate attacks since the integration of the semantic information has been demonstrated to not improve the adversary efficacy as discussed in related studies [16, 21].

4.3 Experiments

Here, we present experimental settings and the discussion of the empirical results.

4.3.1 Settings

In this section, we describe the the experimental evaluation and provide details necessary to reproduce the experiments. First, we introduce the two real-world datasets used in recommendation scenarios (Section 4.3.1), as well the process carried out to extract, select and filter the semantic information obtained from the KG (Section 4.3.1 to 4.3.1). Afterward, we present the four collaborative filtering (CF) recommendation models tested against the proposed attacks (Section 4.3.1). Finally, we detail the evaluation metrics and the experimental setting used for the experimental evaluation (Section 4.3.1 and 4.3.1).

Dataset

We test the proposed shilling attack approach on two recommendation datasets: `LibraryThing` and `Yahoo!Movies`.

- `LibraryThing` [87] is a popular dataset whose interactions originate from the LibraryThing website [3], a social cataloging web application. The dataset contains

[3]https://www.librarything.com/

user-item rating scores ranging from a minimum of 1 to a maximum of 10. As presented by Anelli et al. [16], we use a reduced version by randomly extracting the 25% of products in the catalog. Furthermore, we apply a 5-core filtering by removing all the users with less than five interactions to focus the study on active users. These users are of adversaries' interest since they could more likely buy the pushed products.

- `Yahoo!Movies` is a recommendation dataset released by `research.yahoo.com` with ratings collected up to November 2003. The dataset also provides mappings to the MovieLens and EachMovie catalogs. The recorded interactions consist of ratings ranging from 1 to 5.

Another motivation for choosing these datasets is the existence of a mapping between the products in the catalogs and `DBpedia` knowledge-base entities. In particular, we use a mapping publicly available [4]. Table 4.2 reports the statistics of both datasets' user-item interaction data, together with the total number of semantic features extracted from both the first and the second hop of the knowledge graph associated with each item. In the following, we describe steps taken for pre-processing and data sanity of the features extracted from a KG.

Feature Extraction. Once the items are semantically reconciled with `DBpedia` entities, we remove the noisy features whose triples contain one of the following predicates:

- `owl:sameAs`
- `dbo:thumbnail`
- `foaf:depiction`
- `prov:wasDerivedFrom`
- `foaf:isPrimaryTopicOf`

The feature's denoising procedure follows the methodology proposed by Anelli et al. [24].

[4] https://github.com/sisinflab/LinkedDatasets

Table 4.3 Selected features in the different settings, either for single and double hops.

	Single hop features						Double hop features					
	Categorical		Ontological		Factual		Categorical		Ontological		Factual	
Dataset	Total	Selected	Total	Selected	Total	Selected	Total	Selected	Total	Selected	Total	Selected
`LibraryThing`	3,890	373	2,090	311	50,039	1,972	9,641	857	3,723	527	4,246,365	252,848
`Yahoo!Movies`	5,555	1,192	3,036	722	97,142	7,690	8,960	1,956	3,105	431	6,685,921	517,211

Feature Selection. To perform the analysis of the class (or type) of semantic features, we implement our proposed semantics-aware attacks by considering three different types of features, i.e., categorical (CS), ontological (OS), and factual (FS), a feature taxonomy commonly adopted in the Semantic Web community [24]. For the semantics-aware attack strategies exploiting single-hop (**1H**) features, we apply the following policies:

- **Categorical-1H**, we use the features with the property `dcterms:subject`;
- **Ontological-1H**, we select the features containing the property `rdf:type`;
- **Factual-1H**, we consider all the features except ontological and categorical features.

In the attacks employing double-hop (**2H**) features, the strategies evolve as described below:

- **Categorical-2H**, we pick up the features with either `dcterms:subject` or `skos:broader` properties;
- **Ontological-2H**, we select the features containing either `rdf-schema:subClassOf` or `owl:equivalentClass` properties;
- **Factual-2H**, we use the features not selected in the previous two classes.

Note that we did not place any domain-specific categorical/ontological feature in the respective lists. To provide a domain-agnostic evaluation, we have treated them as factual features.

Feature Filtering. In this chapter, we aim to study the attack performance differences up to the first and second hop. Addressing this goal, we obtain millions of features for both `LibraryThing` and `Yahoo!Movies` as reported in the last two columns of Table 4.2. Measuring semantic similarities across the item catalog would quickly become unfeasible. However, some features only occur once and provide no useful

informative or collaborative information. Therefore, we decided to drop off irrelevant features following the filtering technique proposed in Di Noia et al. [87], Paulheim and Fürnkranz [180]. In detail, we removed all the features with more than 99.74% of missing values and distinct values. Table 4.3 shows the remaining features' statistics after applying all the extraction, selection, and filtering process.

Recommender Models

In this chapter, we test our attack proposal against four baseline collaborative recommendation systems: User-kNN, Item-kNN, Matrix Factorization, and Neural Matrix Factorization. The first two approaches belong to **memory-based** CF, while the next two are model-based CF, thus providing us an overall picture of different recommendation models' performance when confronted with shilling attacks.

- **User-kNN** is presented in Section 3.3.1. We use the *Pearson Correlation* as the distance metric $dist(\cdot)$ as suggested by Candillier et al. [52]. The size of the neighborhood, k, is set to 40.

- **Item-kNN** is presented in Section 3.3.1. Similar to User-kNN, we use the *Pearson Correlation* to implement the distance function $dist(\cdot)$ and set k the dimension of the considered neighborhood 40.

The third and fourth recommendation systems are representative of **model-based** collaborative recommenders. In particular, matrix factorization is the baseline recommender representing the class of linear latent factor models, while neural matrix factorization represents the class of non-linear models.

- **Matrix Factorization (MF)** is defined in Section 2.1.1. Following the learning settings defined in [127], we set the size of latent vectors h to 100.

- **Neural Matrix Factorization (NeuMF)** [116] is one of the most representative recommendation model that exploits deep neural networks to estimate unknown user-item preference scores [242]. NeuMF makes use of both the linearity of MF and the non-linearity of neural layers to improve the learning capability of the model. Unlike MF, the estimated score for a $user - item$ pair of the neural network, $\hat{s}_{ui}$, is the output of a deep neural network whose input is the combination of the MF layer and the neural network layer. The latter concatenates the user ($\mathbf{p}_u$)and the item ($\mathbf{q}_i$) embeddings. Let $\Phi(\cdot)$ be the transformation

function of the deep neural network defined as $\Phi(x) := \mathbb{R}^{dim(x))} \rightarrow \mathbb{R}^{out\_dim}$, then the score is predicted as follows:

$$\begin{aligned} \phi^{GMF} &= \mathbf{p}_u \odot \mathbf{q}_i \\ \phi^{MLP} &= \Phi([\mathbf{p}_u, \mathbf{q}_i]) \\ \hat{s}_{ui} &= \sigma(H^T \begin{bmatrix} \phi^{GMF} \\ \phi^{MLP} \end{bmatrix}) \end{aligned} \tag{4.6}$$

where $\odot$ denotes the element-wise product of vectors, whereas σ and H denote the activation function and edge weights of the output layer, respectively. In Equation (4.6), $\mathbf{q}_i \in \mathbb{R}^{h_1}$ and $\mathbf{p}_u \in \mathbb{R}^{h_2}$ are the latent representations of user u and item i that are concatenated via the function $[\cdot]$, i.e., the input of the deep neural network. We set $h_1 = h_2 = 16$ as suggested by He et al. [116]. The vector resulting from the concatenation of $\mathbf{p}_u$ and $\mathbf{q}_i$ is fed into a deep neural network composed by 4 fully connected dense layers with {64, 32, 16, 8} hidden units, respectively. During the training, we insert a dropout pre-layer for each of the four layers with a dropout rate equal to 0.1.

Evaluation Metrics

To perform the evaluation of the proposed attack we use the HR@K($\mathcal{I}_T$,$\mathcal{U}_T$) and PS($\mathcal{I}_T$,$\mathcal{U}_T$) defined in Section 2.3.3 (see Definitions 19 and 20).

Evaluation Protocol

To investigate the impact of the proposed attack strategies, we perform 360 experiments for each pair of a dataset and the number of extracted hops, totaling 1,440 experiments. Following the evaluation procedure used in Mobasher et al. [161], Lam and Riedl [143], we generate the list of recommendations for each recommendation model before executing the attack. After having measured the position and predicted score for each target item-user pair, we simulated the attack. First, we craft and add shilling profiles to the data following the baseline attack strategies. The HR@K($\mathcal{I}_T$,$\mathcal{U}_T$) and PS($\mathcal{I}_T$,$\mathcal{U}_T$) results extracted from the model's training on the poisoned data constitute the baselines to compare with semantic attack strategies. Then, we evaluate the same metrics on the recommendation results generated on the data poisoned by the fake profiles crafted with the proposed strategy (details in Section 4.2). Note that we evaluate the semantic strategies considering a scenario where the adversary's goal is to *push* a target item/product. In particular, we perform each one of the 360 experiments

on 50 randomly selected items in the dataset. Furthermore, we perform each attack using three different amounts of injected shilling profiles: $1\%, 2.5\%$, and 5% of the total number of users, as adopted in [16, 80, 163]. Regarding the relatedness measures, we set the $\alpha = 0.25$ and the t-path length to 10 for both metrics. To grant the results' reproducibility, the experimented datasets and the code are publicly available.[5]

4.3.2 Results and Discussion

Since the study analyzed several aspects, the investigations can be summarized to address the following research questions to provide a general overview:

RQ1 Can relatedness-based measures along with public available semantic information be employed to develop more effective shilling attack strategies against recommendation models?

RQ2 Can we assess which is the most impactful type of semantic information?

RQ3 Is multiple hops exploration of a knowledge graph more effective than single-hop exploration to create coherent fake profiles?

RQ4 What are the recommendation algorithms that suffer more for semantics-aware attacks?

All the results are computed for top-10 recommendation, i.e., $K = 10$. To avoid redundancy, we will refer to $HR@10(\mathcal{I}_T, \mathcal{U}_T)$ with HR in the rest of the chapter.

Tables 4.4 and 4.5 report the HR values measured for each of the 360 attack combinations experimented on the `Yahoo!Movies` and the `LibraryThing` datasets, respectively. Across the next sections, we identify an attack combination using the format <dataset, hops, recommendation model, attack strategy, feature type, similarity measures, attack granularity>. For example, <`Yahoo!Movies`, 1H, User-kNN, Average, Categorical, Katz, 1%> indicates an experiment on the `Yahoo!Movies` dataset when the adversary uses the average semantics-aware strategy against a User-kNN recommendation model. Here, the semantic features are the categorical ones extracted from the first hop and exploited by the adversary by measuring the Katz-relatedness between each item in the catalog. Finally, 1% shows the percentage fraction of fake profiles added into the training data.

By comparing the results across the two datasets, the first observation is that the results obtained on the `Yahoo!Movies` dataset (Table 4.5) are more indicative of

[5]https://github.com/sisinflab/SAShA-against-CFRS

Table 4.4 Hit Ratio (HR) result values evaluated on top-10 recommendation lists for the `LibraryThing` dataset. We use the following notations: R (Random), A (Average), and B (Bandwagon).

			User-kNN			**Item-kNN**			**MF**			**NeuMF**		
Attack	**Feature**	**Sim.**	1	2.5	5	1	2.5	5	1	2.5	5	1	2.5	5
R	Baseline		*.0736*	*.1570*	*.2301*	*.2885*	*.4588*	*.5590*	*.7660*	*.8987*	*.9419*	*.0612*	*.1130*	*.2216*
	Cat.	Cosine	.0745	.1576	.2311	.2804	.4575	.5687	.7837	.9014	**.9439**	.0802	.1324	.1653
		Katz	.0808	.1698	.2441	.2862	.4610	.5691	.7885	.9021	.9418	.0808	.1105	.1812
		Excl.	**.0816**	**.1703**	**.2456**	**.2915**	.4635	.5707	.7897	.8993	.9427	.0886	.1479	**.2417**
	Ont.	Cosine	.0709	.1503	.2252	.2748	.4483	.5634	.7720	.8979	.9423	.0561	**.1493**	.1926
		Katz	.0774	.1622	.2355	.2837	.4592	.5670	.7845	.9021	.9416	.0751	.1392	.1857
		Excl.	.0766	.1619	.2349	.2848	.4602	.5686	.7846	.9010	.9433	**.1091**	.0999	.2240
	Fact.	Cosine	.0740	.1558	.2280	.2786	.4528	.5642	.7835	.9023	.9419	.0676	.1009	.1285
		Katz	.0760	.1591	.2319	.2823	.4570	.5662	.7839	.9015	.9417	.0685	.1366	.1823
		Excl.	.0793	.1672	.2425	.2890	**.4646**	**.5722**	**.7888**	**.9029**	.9434	.0921	.1034	.2143
A	Baseline		*.0857*	*.1994*	*.2863*	*.3170*	*.5085*	*.6070*	*.8043*	*.9140*	*.9500*	*.0416*	*.0670*	*.1362*
	Cat.	Cosine	.0864	.1967	.2823	.3060	.5115	.6202	.8128	.9127	.9502	.0634	.0950	.1316
		Katz	.0940	**.2094**	**.2922**	.3136	.5133	.6136	.8149	.9132	.9486	.0630	.1031	.1119
		Excl.	**.0941**	.2074	.2888	**.3185**	**.5142**	.6142	.8165	.9128	.9502	.0482	.0586	.1548
	Ont.	Cosine	.0849	.1954	.2805	.3073	.5126	**.6207**	.8114	.9163	**.9509**	**.0906**	**.1248**	**.1569**
		Katz	.0898	.2021	.2845	.3096	.5107	.6143	.8168	.9135	.9491	.0816	.1171	.1108
		Excl.	.0890	.2020	.2842	.3119	.5119	.6165	.8121	.9145	.9489	.0285	.0599	.0947
	Fact.	Cosine	.0868	.1989	.2806	.3073	.5112	.6185	.8163	**.9166**	.9471	.0362	.0851	.1222
		Katz	.0892	.2016	.2844	.3098	.5110	.6158	**.8189**	.9139	.9473	.0588	.0849	.1040
		Excl.	.0912	.2049	.2872	.3152	.5131	.6131	.8166	.9138	.9482	.0502	.0746	.0882
B	Baseline		*.0817*	*.1319*	*.1881*	*.2640*	*.3834*	*.4694*	*.6000*	*.7656*	*.8435*	*.0100*	*.0105*	*.0061*
	Cat.	Cosine	.0763	.1234	.1752	.2641	.3801	.4632	.5918	**.7661**	.8429	**.0107**	.0077	.0074
		Katz	.0794	.1266	.1800	**.2647**	.3821	.4648	.5896	.7596	.8422	.0103	.0080	**.0094**
		Excl.	.0758	.1227	.1745	.2640	.3818	.4646	.5835	.7590	.8435	.0067	.0054	.0068
	Ont.	Cosine	.0758	.1227	.1745	.2626	.3798	.4637	.5904	.7619	.8433	.0064	.0056	.0049
		Katz	.0792	.1257	.1779	.2636	.3802	.4637	.5820	.7642	**.8447**	.0051	.0027	.0077
		Excl.	.0776	.1249	.1770	.2633	.3815	.4643	.5979	.7611	.8413	.0057	.0047	.0052
	Fact.	Cosine	.0738	.1190	.1714	.2632	.3784	.4623	**.6001**	.7634	.8408	.0057	.0044	.0063
		Katz	.0776	.1239	.1771	.2641	.3801	.4630	.5833	.7602	.8415	.0026	.0083	.0036
		Excl.	.0792	.1272	.1796	.2638	.3813	.4642	.5948	.7590	.8405	.0051	.0054	.0227

We underline the results with a p-value greater than 0.05 *using a paired-t-test statistical significance test.*

attacks' effectiveness independently of the attack strategy, the number of injected profiles, and recommender models, confirming the findings in our previous work, Anelli et al. [16]. One plausible explanation for this behavior is the differences in dataset characteristics, e.g., the data sparsity, that has been showing impacting shilling attacks' performance as verified by Deldjoo et al. [76].

Furthermore, Table 4.4 also confirmed the semantics-aware strategy's efficacy over the baseline, either for the average and random attacks. For instance, the semantic strategies outperformed all the <`LibraryThing`, 1H, Random> and <`LibraryThing`, 1H, Average> baseline attacks independently of the recommender model and the size of attacks. However, it is worth mentioning that, differently from the results on `Yahoo!Movies`, on <`LibraryThing`, 1H, Bandwagon>, the baseline attack's effectiveness did not improve. This behavior might be linked with semantic information extracted from the KG and the attack strategy itself. Since a bandwagon attack builds profiles by filling the 50% of the profile with the most *popular* items, it might make the semantic

Table 4.5 Hit Ratio (HR) result values evaluated on top-10 recommendation lists for the `Yahoo!Movies` dataset. We use the following notations: R (Random), A (Average), and B (Bandwagon).

			User-kNN			**Item-kNN**			**MF**			**NeuMF**		
Attack	**Feature**	**Sim.**	1	2.5	5	1	2.5	5	1	2.5	5	1	2.5	5
R	Baseline		*.1927*	*.3624*	*.4461*	*.3260*	*.5099*	*.6011*	*.4108*	*.5857*	*.7043*	*.0247*	*.0221*	*.0700*
	Cat.	Cosine	.1869	.3512	.4277	.3163	.4980	.5886	.4084	.5720	.6648	.0018	.0127	.0464
		Katz	.1912	.3725	.4559	.3429	.5270	.6098	.4244	.6029	.7049	.0223	.0317	**.0891**
		Excl.	.1968	.3712	.4533	.3394	.5233	.6072	.4272	.6011	.7023	.0171	.0516	.0544
	Ont.	Cosine	.1730	.3353	.4163	.2994	.4793	.5726	.3916	.5513	.6407	.0030	.0051	.0118
		Katz	.1766	.3547	.4337	.3224	.5046	.5904	.4029	.5698	.6638	.0106	.0191	.0386
		Excl.	**.2101**	**.3898**	**.4706**	.3532	**.5442**	**.6243**	**.4450**	**.6328**	**.7376**	.0242	**.0567**	.0515
	Fact.	Cosine	.1881	.3501	.4289	.3149	.4933	.5840	.4087	.5665	.6590	.0188	.0115	.0365
		Katz	.2094	.3869	.4703	**.3545**	.5398	.6213	.4442	.6272	.7371	**.0368**	.0507	.0269
		Excl.	.2055	.3799	.4632	.3479	.5317	.6178	.4361	.6142	.7187	.0176	.0402	.0430
A	Baseline		*.2293*	*.4117*	*.4918*	*.3758*	*.5759*	*.6564*	*.4900*	*.6824*	*.7849*	*.0033*	*.0044*	*.0236*
	Cat.	Cosine	.2581	**.4296**	.4972	.3955	**.5953**	.6689	.5326	.7255	.8076	.0017	**.0383**	.0029
		Katz	.2319	.4142	.4917	.3882	.5773	.6542	.4889	.6777	.7716	.0015	.0064	**.0272**
		Excl.	.2277	.4026	.4845	.3752	.5698	.6493	.4813	.6658	.7624	.0064	.0014	.0087
	Ont.	Cosine	**.2584**	.4264	.4953	**.4019**	.5952	**.6704**	**.5457**	**.7315**	**.8128**	.0043	.0018	.0111
		Katz	.2406	.4209	.4964	.3940	.5877	.6615	.5131	.7093	.7950	.0040	.0022	.0098
		Excl.	.2196	.3965	.4771	.3623	.5531	.6337	.4552	.6401	.7347	**.0099**	.0348	.0205
	Fact.	Cosine	.2573	.4290	**.4960**	.3882	.5884	.6634	.5353	.7256	.8009	.0026	.0055	.0054
		Katz	.2293	.4101	.4910	.3736	.5608	.6414	.4746	.6559	.7511	.0073	.0047	.0231
		Excl.	.2311	.4075	.4894	.3706	.5661	.6467	.4809	.6661	.7602	.0042	.0070	.0194
B	Baseline		*.0996*	*.2418*	*.3556*	*.2427*	*.3764*	*.4691*	*.2357*	*.3606*	*.4320*	*.0010*	*.0026*	*.0025*
	Cat.	Cosine	.1020	.2544	**.3634**	.2453	.3831	.4748	.2536	.3909	.4662	.0010	.0208	.0010
		Katz	.0981	.2412	.3495	.2383	.3676	.4546	.2300	.3540	.4248	.0017	.0022	.0077
		Excl.	.0926	.2357	.3476	.2378	.3670	.4562	.2248	.3472	.4150	.0009	**.0094**	.0026
	Ont.	Cosine	.1039	**.2632**	.3606	.2460	**.3853**	**.4786**	**.2726**	**.4080**	**.4798**	.0045	.0060	.0009
		Katz	.0958	.2476	.3528	.2412	.3754	.4652	.2253	.3602	.4376	.0009	.0023	.0012
		Excl.	.0941	.2227	.3346	.2289	.3528	.4402	.2092	.3191	.3885	.0030	.0022	.0054
	Fact.	Cosine	**.1050**	.2562	.3614	**.2476**	.3814	.4734	.2506	.3890	.4625	.0133	.0043	.0004
		Katz	.0930	.2302	.3460	.2295	.3569	.4461	.2178	.3399	.4064	**.0255**	.0028	**.0115**
		Excl.	.0926	.2360	.3515	.2345	.3616	.4504	.2309	.3446	.4137	.0023	.0012	.0014

We underline the results with a p-value greater than 0.05 *using a paired-t-test statistical significance test.*

strategy that identifies the informative filler items ineffective. These new insights show the nuances captured by our proposed semantics-aware strategies for enriching state-of-the-art shilling attack methods against CF models.

Below, we provide a more in-depth discussion about the impact of several factors involved in the design space of the proposed semantics-aware shilling attacks against CF models. They include the effect of the feature type extracted from the KG, i.e., CS, OS, or FS, the semantic similarity/relatedness between the target item and the items in the catalog, and the hop depth described in detail in Section 4.3.1. Our goal is to answer the research questions provided in Section 4.1 along with these directions.

Impact of Relatedness-based Measures and Semantic Data (RQ1)

The first research question is intrinsically the most important one. Given the extent of experiments carried out in the experimental section, it could be hard to decipher

this information at first glance. Thus, in this section, we try to decode the insights obtained from the experimental results along the experimental directions outlined above. Let us consider the experiments on `LibraryThing`. We can observe that the adoption of graph-based relatedness generally leads to an attack efficacy improvement over the baseline, which adopts the cosine similarity metric. For instance, the random attack (where the attacker does not have system knowledge) primarily benefits from the topological information. The general observation here is that in most experimental cases, the adoption of relatedness-based semantic information leads to improvement of the attacks' effectiveness. We may observe the same behavior for the `Yahoo!Movies` dataset in Table 4.5, in which the HR for <1H, User-kNN, Random, Categorical, Katz> is 10% better than the baseline, i.e., 0.3725 vs. 0.3512.

Beyond random attacks, we can observe some general trends also for informed attacks. In detail, Table 4.4 (`LibraryThing`), we note that categorical information improves both User-kNN and Item-kNN. It is worth noticing that the same consideration does not hold for latent factor-based models. MF and NeuMF suit better cosine vector similarity. This phenomenon is probably due to the significant difference in how the two recommendation families exploit the additional information. Finally, we can focus on the Bandwagon attack. In that case, the attack already exploits the most influential knowledge source for collaborative filtering algorithms: popularity. It follows that the integration with other knowledge sources, e.g., KGs, does not provide any significant improvement. However, the influence of popularity is so high in this attack that the final recommendation lists are subject to a strong popularity bias [1]. Indeed, adding fake profiles with the maximum ratings, e.g., 5 in `Yahoo!Movies` and 10 in `LibraryThing`, placed on the most popular/rated items that will form the $\mathcal{I}_S$ (see Tables 2.2 and 4.1) will amplify, even more, the probability that these items will be recommended in the highest positions of top-K recommendation lists making ineffective the adversaries' pushing goal toward the target items.

As a consequence, it even prevents the attacked recommendation system from suggesting the target item. All the experimental datasets and all the recommendation models clearly show this effect.

Another aspect that we want to underline is that increasing the number of fake profiles injected into the systems unleashes the potential of different semantic knowledge types. For instance, in the <`LibraryThing`, Average, MF> setting with 1% injected fake profiles, we observe the best results with Factual knowledge and *Katz* centrality, while, with 2%, the best results are with Factual knowledge and cosine similarity. Finally, with 5%, the best results come with Ontological knowledge and cosine similarity. This

behavior suggests that the graph-based similarities have a significant impact even in a very sparse scenario. In contrast, with the increase of fake profiles, the cosine similarity starts leveraging interesting correlations. On the other dimension, factual information is massive by nature, and it is crucial in sparse scenarios. However, when the number of fake profiles increases, the knowledge at a higher level of abstraction (Categorical and Ontological) finds its way to improve the attack efficacy further.

Impact of Factual, Ontological, and Categorical Data (RQ2)

The following essential aspect to investigate is the combined impact of semantic knowledge type and relatedness measure. In detail, we believe this is a straightforward natural evolution of RQ2. We start focusing on Categorical knowledge. The experiments on `LibraryThing` show that Exclusivity is probably the relatedness that best suits this information type. However, the results are not that clear for the `Yahoo!Movies` dataset. This behavior suggests that semantic information type and relatedness are not the only members of the equation. Indeed, the extension and the quality of the item descriptions seem to have a role. Afterward, we can focus on Ontological information. Here, we can draw a general consideration since, for both datasets, it is the cosine similarity metric that leads to the best results. Lastly, Factual information respects all the general remarks we have drawn before, showing that the relatedness is a better source of adversaries' knowledge to perform more effective attacks.

In detail, we found that with low-knowledge attacks, the best relatedness is *Exclusivity* for `LibraryThing` and *Katz* for `Yahoo!Movies`. With informed attacks, the best relatedness metric is the cosine similarity. However, for the sake of electing a similarity that better suits Factual information, we can note that *Exclusivity* generally leads to better results with `LibraryThing`.

Analysis of KG's Hops (RQ3)

The subsequent analysis focuses on the impact of the 1-hop and 2-hops of the KG exploration. To support this analysis, we have prepared the summary table. Table 4.6 firstly, shows the average variation of attack efficacy passing from the adoption of single-hop extracted features to the double-hop extraction for `LibraryThing` and `Yahoo!Movies`. Regarding `Yahoo!Movies`, the first and foremost consideration we can draw is that graph-based relatedness measures seem to have no positive impact when exploiting a double-hop exploration. However, it can be observed that those relatedness metrics already achieved impressive results with the first-hop exploration. Hence, further improving the performance is somehow challenging. Indeed, in most

Table 4.6 Variation of Hit Ratio (HR) when using the features extracted from the second hop with respect to the first hop for `LibraryThing` and `Yahoo!Movies`.

			LibraryThing				Yahoo!Movies			
Attack	**Feature**	**Sim.**	**U-kNN**	**I-kNN**	**MF**	**NeuMF**	**U-kNN**	**I-kNN**	**MF**	**NeuMF**
Random	Cat.	Cos.	-1.28	-1.63	-0.70	-20.07	-0.03	-0.01	-0.01	1.57
		Katz	-0.77	2.05	-0.20	-6.05	-0.11	-0.10	-0.06	-0.47
		Exc.	-2.12	0.14	-0.26	-21.09	-0.05	-0.04	-0.02	0.08
	Ont.	Cos.	1.97	0.64	0.35	13.45	0.16	0.12	0.10	1.31
		Katz	-3.00	-0.24	0.10	-38.28	-0.07	-0.07	-0.04	-0.29
		Exc.	-4.57	-1.92	-0.47	-46.85	-0.13	-0.09	-0.07	-0.66
	Fact.	Cos.	-0.64	-0.62	-0.11	46.94	-0.01	0.02	0.01	-0.62
		Katz	0.93	2.60	0.07	56.47	-0.12	-0.09	-0.07	-0.73
		Exc.	-0.33	0.25	-0.39	-29.80	-0.16	-0.11	-0.08	-0.21
Average	Cat.	Cos.	-0.87	-0.86	-0.21	-17.66	-0.03	0.00	-0.01	0.67
		Katz	0.07	2.13	0.02	36.36	0.03	-0.03	0.05	3.81
		Exc.	-1.82	-0.09	-0.22	52.37	0.02	-0.02	0.03	-0.69
	Ont.	Cos.	0.47	-0.05	0.22	-8.44	-0.14	-0.12	-0.17	-0.19
		Katz	-3.92	-0.82	-0.52	-70.51	0.07	0.00	0.06	2.94
		Exc.	-4.49	-2.26	0.32	152.52	0.07	0.02	0.06	-0.77
	Fact.	Cos.	-0.19	0.29	0.06	123.56	-0.04	0.00	-0.04	0.22
		Katz	0.64	1.73	-0.28	13.12	0.01	-0.02	0.04	-0.75
		Exc.	0.53	0.87	-0.33	-2.11	0.06	0.03	0.09	-0.17
BandWagon	Cat.	Cos.	-0.02	-0.55	-0.42	-51.24	-0.03	0.00	0.02	-0.01
		Katz	-1.93	-1.01	-0.04	-68.96	-0.06	0.02	0.00	8.87
		Exc.	3.25	-0.32	0.07	36.58	0.02	-0.02	0.05	0.07
	Ont.	Cos.	-1.37	-0.10	0.16	49.05	-0.14	-0.08	-0.20	-0.62
		Katz	-5.69	-0.18	2.05	-9.28	0.01	-0.01	0.10	0.78
		Exc.	-2.37	-0.45	-0.55	-35.24	-0.02	0.02	0.10	0.61
	Fact.	Cos.	1.80	-0.14	-0.32	5.18	-0.07	-0.02	-0.02	-0.91
		Katz	1.57	-0.45	1.00	190.44	0.02	0.05	0.07	-0.90
		Exc.	-1.57	-0.61	-1.52	140.00	0.07	0.03	0.08	-0.17

cases, we can observe a minimal variation for the double-hop performance. However, in some cases, the attacks witness a more significant decrease, probably due to the injection of some noisy and loosely-related second-hop features. In general, given the high performance achieved with a single-hop exploration, it seems that it is not worth exploring the second-hop, and thus increasing the computational complexity and introducing the new challenge of loosely-related second-hop features. Beyond graph-based relatedness, we observe that cosine vector similarity almost always shows an improvement when considering second-hop features (particularly with Ontological and Factual information). Finally, we have to observe that, even here, the NeuMF model does not benefit from this new information.

Table 4.6 also shows the average attack efficacy variation for `LibraryThing`. Here, some previously described behaviors are even more evident. In detail, we note that the cosine similarity takes advantage of the second-hop information. In this case, we can

also observe *Katz*'s improvement, suggesting that this metric did not have unleashed its full potential with only the first-hop features. Finally, in some cases, the second-hop information also improves informed attacks (reaching a peak of 53% improvement for <Average, Factual, *Exclusivity*>), confirming a less evident trend we found with `Yahoo!Movies`.

Analysis of RS Vulnerability (RQ4)

The last discussion analyzes the efficacy of the semantic attacks on the different recommendation families. Since the neighborhood-based models directly exploit a similarity to compute the recommendation lists, they are the privileged victim models to alter the recommendation performance effectively. Indeed, both user-based and item-based schemes heavily suffer from semantics-aware shilling attacks. The publicly available semantic information can help the attacker crafting impactful fake profiles even in the case of a complete lack of information about the system, e.g., `SAShA`-Random results. Even though latent factor models seem to be more robust to the attacks, semantic attacks improved the attacker's performance. Finally, the most robust model seems to be NeuMF. This result is probably due to the non-linearity of NeuMF that helps the model avoid learning from the pretended profiles. In detail, the neural network may learn more sophisticated correlations that the other models do not capture. We believe that this ability deserves specific further investigation since it may lead to developing a new line of research on Deep Learning-based semantics-aware attacks that might exploit non-linear item-item similarities to build more impactful attack methods.

4.4 Related Work

All of us have witnessed the astonishing performance of recommendation systems. However, few know that, often, the recommendation algorithms struggle to optimize the model. Despite the number of transactions being massive, the number of per-user interactions is usually very scarce. Over the years, the recommendation system designers relied on additional sources of information to overcome this limitation. Nowadays, modern RSs exploit various side information such as metadata (e.g., tags, reviews) [169], social connections [34], image and audio signal features [75], and users-items contextual data [10] to build more in-domain [107] (i.e., domain-dependent), cross-domain [96], or context-aware [129] recommendation models. Among the diverse information sources, what is, likely, the most relevant source is Knowledge Graphs (KGs). A KG is a

heterogeneous network that encodes multiple relationships, edges, nodes, and links items at high-level relationships, making them a strong item representation technique. Thanks to the heterogeneous domains that KGs cover, the design of knowledge-based recommendation systems has arisen as a specific research field of its own in the community of RSs, usually referred to by Knowledge-aware Recommender Systems (KaRS [22]). This research community combines the most advanced machine learning techniques with state-of-the-art knowledge representation paradigms. This blending of skills and ideas has generated several advancements in the recommendation [24], knowledge completion [111], preference elicitation [30], user modeling [219] research, and thus produced a vast literature.

A comprehensive review of the field would require a separate and specific paper; however, we can still provide an overview of the most advanced (or particularly representative) contributions. To help the reader orient herself in the literature, we follow three distinct lines: impacted research fields, recommendation techniques, and data sources. In recent years, the Knowledge-aware Recommender Systems have been particularly impactful for several research domains:

- **KG-embeddings** [176, 147, 166], where the latent representation of semantic knowledge enables novel and diverse applications;
- **Hybrid Collaborative/Content-based recommendation** [147, 24], exploiting the KG information to suffice the lack of collaborative information and to improve the performance;
- **Knowledge-completion**, **link-prediction**, and **knowledge-discovery** [111, 47], where the topology of the knowledge graph and the graph embeddings helped to improve the overall quality of the knowledge base;
- **Knowledge-transfer, cross-domain recommendation** [240, 96], where the KGs allow to find semantic similarities between different domains;
- **Interpretable/Explainable-recommendation** [11, 25, 231], with KG being a backbone for understanding the recommendation model and providing human-like explanations
- **User Modeling** [219, 172, 132], since the resource descriptions can drive the construction of the user profile;

- **Graph-based recommendation** [194, 224, 198, 220], where the topology-based techniques have met the semantics of the edges/relations, and the ontological classification of nodes (classes);
- **The cold-start problem** [165, 96], since the KGs can overcome the lack of collaborative information;
- **The content-based recommendation** [26] that solely relies on KG and still produces high-quality recommendations.

All the former advances have been shown to enhance the recommendation quality or the overall user experience. Although the algorithms differ on many levels, we can still classify recommendation techniques into two broad approaches:

- **Path-based** methods [194, 224, 198, 87], which employ paths and meta-paths to estimate the user-item similarities or the nearest items;
- **KG embedding-based** techniques [194, 166, 24], which leverage KG embeddings (usually obtained through matrix factorization or neural network encoding) for items' representation.

Finally, we focus on the Knowledge Graphs data sources. The availability of a myriad of KGs is a definite advantage of Knowledge-aware Recommender Systems. Thanks to the Linked Data initiative, today, we can benefit from 1,483 KGs connected in the so-called Linked Open Data Cloud[6]. KGs can be general-purpose, or domain-specific like Academia/Industry DynAmics (AIDA) [31]. However, most contributions concentrate on a short-list of KGs with a peculiar characteristic: being an encyclopedic KG. Those KGs share the same ontology and the same schema across multiple domains, giving access to huge knowledge at the exact development cost required for a single domain. The most appreciated KGs of this special class undoubtedly are `DBpedia` [145], Wikidata [217], Yago [207] (the 4th release [210] also supports RDF* [110]), FreeBase [45], Satori[7][8] [151], Google's Knowledge Graph[9], Knowledge Vault [88], Bio2RDF [38].

Despite the extensive use of KGs in recommendation tasks, we have not identified any malicious use of these vast sources of additional data. Indeed, a typical characteristic of the previous literature on shilling attack strategies is that they usually target the

[6]https://lod-cloud.net/datasets
[7]https://searchengineland.com/library/bing/bing-satori
[8]https://blogs.bing.com/search/2013/03/21/understand-your-world-with-bing
[9]https://blog.google/products/search/introducing-knowledge-graph-things-not/

relations between users, and items, based on similarities scores estimated on their past feedback (e.g., ratings). However, these strategies do not consider the possibility of exploiting publicly available semantic information to gain more information on the semantic similarities between the items available in the RS catalog. Indeed, considering that product or service providers' catalogs are freely accessible to everyone, this chapter has presented a novel attack strategy that exploits a freely accessible knowledge graph (`DBpedia`) to assess if attacks based on semantic similarities between items are more effective than baseline versions that rely only on users' preference scores.

4.5 Summary

This chapter shows how the adoption of structured and freely accessible knowledge (i.e., Linked Open Data repositories) further improves malicious agents' ability to attack a recommendation platform. Knowledge Graphs have already extensively shown that they help build more accurate recommendation systems. However, this technical study is one of the first attempts to exploit the external knowledge to alleviate the attacker's lack of system knowledge. Starting from the state-of-the-art shilling attacks (where the attacker injects fake profiles into the platform to alter the final recommendations), the chapter proposed a broad spectrum of semantics-aware shilling attacks (`SAShA`). To study and test these attacks' efficacy, we have investigated the impact of graph-based metrics (*Katz* centrality and *Exclusivity*-based relatedness), semantic information type, and Knowledge Graph exploration depth. We have analyzed the attack efficacy along each dimension considering three recommendation families: neighborhood-based, latent factor models, and Neural Network-based recommendations systems, totaling $1,440$ experiments. The extensive experimental evaluation has taught us several important lessons.

- The adoption of structured knowledge generally improves by a large margin the attacker's performance.
- The graph-based metrics can efficiently deal with very sparse scenarios, capturing similarities that are otherwise imperceptible.
- The type of semantic information to feed the attacking system has a significant function in enhancing the adversaries' effectiveness. With a few items/entities, the massive factual information has an important role, but as the number of involved entities grows, more structured information (i.e., categorical and ontological information) leads to better results.

- The single-hop exploration is already sufficient to outperform the semantics-unaware techniques, and the second-hop information does not introduce significant further improvements.
- RSs relying on similarity-based algorithms and classical factorization methods heavily suffer from semantic attacks, which perfectly suffice the lack of user interaction knowledge. At the same time, Neural Network-based ones are the sole techniques shown to be more robust, probably thanks to the model's non-linearity.

The robustness of neural models suggests that there is still room for improvements for the semantics-aware attacks to be investigated in future deep learning-based semantic attack proposals. Then, this research direction could be an initial investigation to design a new class of semantics-aware recommendation systems that will be robust to all these advanced attacks.

Chapter 5

Training Time Adversarial Attacks and Defenses on Multimedia RSs

Can an adversary poison the data of multimedia recommender systems with adversarial samples? Do adversarial perturbations of product images confuse multimedia recommenders? Can we protect the model integrity?

Deep learning classifiers are hugely vulnerable to adversarial examples, and their existence raised cybersecurity concerns in many tasks, emphasizing malware detection, computer vision, and speech recognition. While there is a considerable effort to investigate attacks and defense strategies in these tasks, only limited work explores the influence of attacks on input data (e.g., images, textual descriptions, audio) used in multimedia recommender systems (MRSs). For instance, visual-based recommenders enhance recommendation performance by integrating users' feedback with the visual features of items' images.

In this chapter, we present several contributions. Firstly, we examine the consequences of applying targeted adversarial attacks against the product images of VRSs with additional empirical verification of their imperceptibility on final users through state-of-the-art offline-visual metrics. After having asses that human-imperceptible image perturbations, defined adversarial samples, are capable of altering the VRSs performance, for example, by pushing (promoting) or nuking (demoting) specific categories of products, we introduce a set of possible defenses. Mainly, we investigate one of the most effective adversarial defense methods, the *adversarial training* (AT). This technique has been demonstrated to enhance the robustness of ML classifiers against adversarial samples by incorporating them into the training process and minimizing an adversarial risk. While AT effectiveness has been tested in supervised learning tasks

(e.g., image classification), we study whether AT can also protect VRSs against images' adversarial perturbation.

The extensive experiments conducted within an experimental framework, named Visual Adversarial Recommender (`VAR`), indicate alarming risks in protecting a VRS through the DNN robustification.

5.1 Introduction

RSs have terrifically taken over online shopping by providing users with personalized recommendations to disentangle the chaotic flood of products on e-commerce platforms. They model the complex preference that consumers exhibit toward items by leveraging a sufficient amount of past behavioral data. Accordingly, in scenarios such as fashion, food, or point-of-interest recommendation, images associated with products can impact the outcomes of purchasing/consumption decisions, as images attract attention, stimulate emotion, and shape users' first impression about products and brands. To extend the expressive power of RSs, visual-based recommender systems (VRSs) have recently merged that attempt to incorporate products' visual appearance of items into the design space of RS models [83]. Given the representational power of deep neural networks (DNNs) in capturing characteristics and semantics of the images, state-of-the-art VRSs often incorporate visual features extracted via a DNN — pre-trained, e.g., VBPR [114] and ACF [61], or learned end-to-end, e.g., DVBPR [134] — and integrate it with a recommendation model (e.g., MF) to better judge the users' interests.

It follows that DNN serves as a core component of many real-world RSs for performing visually aware recommendation tasks. However, as introduced in Chapter 2, recent studies have demonstrated that adversaries can modify the classification behavior of a trained neural classifier by attaching human-imperceptible adversarial noise on inputs at prediction time [208]. The famous example in the CV domain on the misclassification of a slightly mutated STOP traffic signal into another one installed on a self-driving car [100] has motivated the need to investigate if and how much VRSs might be beatable by adversaries. Indeed, while there is now a sizable body of work proposing different attack and defense strategies in an adversarial setting, namely FGSM [101], PGD [155], and Carlini & Wagner [57] (for the attacks), and Adversarial Training [101], Free Adversarial Training [197], and Defensive Distillation [178] (on the defensive side), we have identified a lack of research on adversarial attacks in the case of multimedia recommenders even though they heavily depend on the benevolence of the product representations extracted from DNNs. The only exception is the work

by Tang et al. [209], that verifies the efficacy of AT in protecting a standard VRS (i.e., VBPR [114]) against adversarial noise applied directly on the image features extracted via ResNet50 [112].

This chapter presents our contribution in the novel proposed motivational situation: a competitor is willing to increase the recommendability of a category of products on an e-commerce platform, e.g., *sandals*, for economic gain. She can achieve this goal by simply uploading adversarially perturbed product images of sandals that are misclassified by the DNN used in the VRS, named image feature extractor (IFE), as a much more popular class, e.g., *running shoes*, allowing sandals to be pushed into recommendation list of more users. This novel adversarial strategy, named Targeted Adversarial Attack against Multimedia Recommender Systems (`BB-TAaMR`), explores attack situations where the adversary's goal is to perturb images of a low recommended category of products (e.g., the 20th most recommended) to be misclassified by the deep classifier towards a target more recommended category (e.g., the 1st/2nd).

The chapter at hand focuses on discovering the unknown vulnerability of VRSs against the poisoning of training data with adversarially perturbed product images constructed to be misclassified by the IFE. In this respect, we propose an empirical framework, named Visual Adversarial Recommendation (`VAR`), to study the efficacy of `BB-TAaMR`, whether and to what extent adversarial training strategies can strengthen IFE's classification performance, thus mitigating the adverse effects of such attacks on the recommendation task and whether the class of VRSs that internally trains the IFE, e.g., DVBPR [134], could be still affected by adversarial samples crafted on a pretrained DNN, e.g., ResNet50, and *transferred* to this end-to-end class of VRSs.

The main contributions of this chapter are summarized as follows:

- an extensive study of adversarial attack methods to implements `BB-TAaMR` in order to break the standard behavior of a VRS to accomplish adversary's desires by guaranteeing the human-imperceptibility of the noise;
- an extensive study of adversarial training (defensive) methods to robustify the visually-aware recommendation performance through the analysis of 156 combinations of three types of IFEs, three attacks, and five VRSs, and three recommendation datasets;
- the proposal of two novel rank-based evaluation metrics, named *category hit ratio* and *category normalized Discounted Cumulative Gain*;

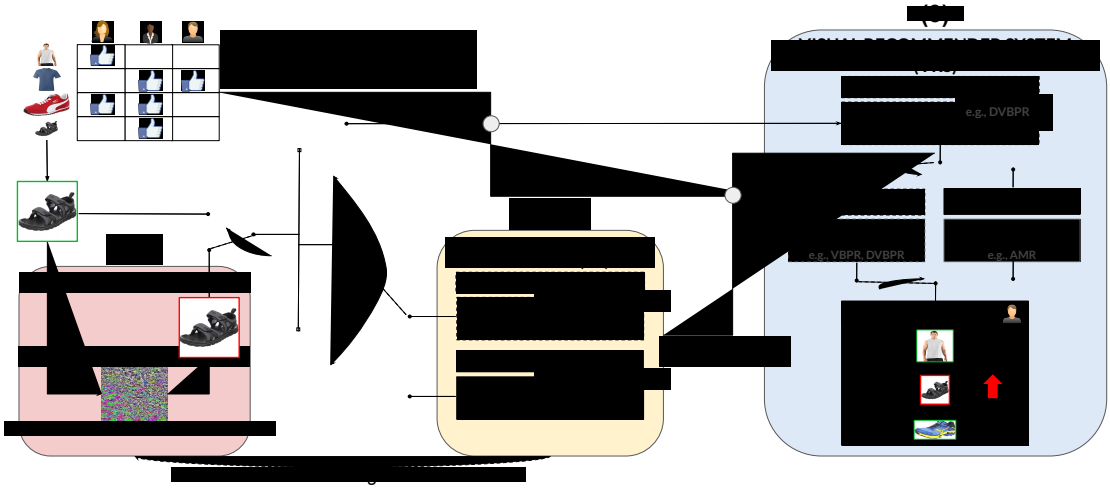


Fig. 5.1 Overview of our VAR framework. (1) an *Adversary* might perturb product images. (2) an *Image Feature Extractor* (IFE) extracts the item visual features. The IFE is implemented either with an external, pre-trained DNN or with a custom DNN within the *Visual Recommender Systems* (VRS). (3) the *Preference Predictor* (PP) from the VRS takes the user-item preference matrix ($\mathcal{R}$) and the visual features to compute the top-K lists. Adversarial training strategies can protect both the external IFE and/or the PP.

- analysis of the variation of global and beyond-accuracy recommendation performance with (and without) defenses to understand to what extent the adversaries in our VAR setting are altering the overall performance of the recommender.

The rest of the chapter is organized as follows. In Section 5.2, we introduce and formalize the proposed framework. In Section 5.3 we describe our experimental settings for study the adversary's capacity in breaking the visual recommender under different constraints. Then, in Section 5.3, we present and discuss empirically evaluate our method. Finally, we review related work in Section 5.4 and summarize the main contributions and future challenges in Section 5.5.

5.2 The Proposed Framework

Here, we describe the VAR components shown in Figure 5.1: *adversary*, *image feature extractor*, and *visual-based recommender system*.

5.2.1 Components

Adversary

To align with the AML literature, we follow the attack —and defense— adversary threat model outlined in [56]. Given all the top-K recommendation lists generated by the VRS, the *adversaries' goal* is to push the items at the bottom of the lists to higher positions. We assume that adversaries are aware of recommendation lists and choose the low-ranked category of items to be pushed (*source*). Then, they select the category of a more recommended item (*target*). Two additional assumptions arise here. The first is that the adversaries have perfect knowledge of the image feature extractor (IFE) used in the VRS, and perturb source images to be misclassified as target ones. The second is that they cannot access the IFE, since it is end-to-end trained along with the VRS, and craft the adversarial samples on another DNN to be transferred to the victim's recommender, i.e., *black-box* attack setting. In our motivating scenario, the adversaries can *poison* the dataset by uploading the adversarially corrupted item images on the VRS-based platform.

Image Feature Extractor (IFE)

The image feature extractor is a deep neural network. Given a set of data samples (x_i, y_i), where x_i is the i-th image and y_i is the one-hot encoded representation of x_i's image category, we define F as a DNN classifier function trained on all (x_i, y_i). Then, we set $F(x_i) = \hat{y}_i$ as the predicted probability vector of x_i belonging to each of all the admissible output classes, and we calculate its predicted class as the index of $\hat{y}_i$ with maximum probability value, and represent it as $F_c(x_i)$. Moreover, assuming an DNN classifier with L-layers, we indicate with $F^{(l)}(x_i)$, $0 \leq l \leq L-1$, the output of the l-th layer of F given the input x_i.

The sample x_i is the image associated with the item $i \in \mathcal{I}$, which may appear in the top-K recommendation list shown to a user. Hence, the IFE is a DNN to extract high-level visual features from x_i. The model can be either *pretrained* on a classification task, i.e., He et al. [112], or a custom network trained *end-to-end* along with the VRS, i.e., Kang et al. [134]. The actual extraction takes place at one of the last layers of the network, i.e., $F^{(e)}(x_i)$, where e refers to the extraction layer. In general, we define this layer output as a three-dimensional vector that will be the input to the VRS. No defense is applied on the custom IFE (see Figure 5.1) used in the end-to-end model (e.g., DVBPR) since defensive approaches only refer to networks trained for the *classification* task. Note that the IFE is a key component in `VAR` since it represents the

connection between the adversary —responsible for the attack— and the preference predictor (PP) used in a VRS.

Visual-based Recommender System (VRS)

In `VAR`, the VRS is the component aimed at addressing the recommendation task. The model takes two inputs: (i) the historical user-item recorded preferences ($\mathcal{R}$), and (ii) the set of item visual features extracted from the pretrained IFE or custom IFE, i.e., DVBPR [134]. Thus, it produces recommendation lists sorted by the preference prediction score evaluated for each user-item pair. Indeed, the VRS preference predictor takes advantage of the pure collaborative filtering source of data, i.e., $\mathcal{R}$, and the high-level multimedia features to unveil user's preferences [114]. In the `VAR` motivating example, the VRS is the final victim of the adversary. For this reason, this chapter focuses on the performance variation of the VRS in attack and defense scenarios.

5.2.2 Evaluation

We perform three levels of investigation, namely: (i) the effectiveness of adversarial attacks in misusing the classification performance of the DNN used as the IFE, (ii) the variation of the accuracy— and beyond-accuracy— recommendation performance, and (iii) the evaluation of consequences for attack and defense mechanisms on the recommendability of the category of items to be pushed.

In AML, several publications focused on quantifying adversarial attacks' success in corrupting the classification performance of a target classifier, i.e., the attack Success Rate (SR) [57]. Similarly, there is vast literature about the accuracy and beyond the accuracy of RSs [191] recommendation metrics. On the other hand, we have observed a lack of literature evaluating adversarial attacks on RSs content data. As a matter of fact, Tang et al. [209] evaluate the effects of untargeted attacks on classical system accuracy metrics, i.e., *Hit Ratio* (HR) and *normalized Discounted Cumulative Gain* (nDCG), while we propose a modified version of HR, named category hit ratio, to evaluate the fraction of adversarially perturbed items in the top-K recommendations, and the normalized Category Discounted Cumulative Gain (nCDCG@K), an updated version of the classical nDCG@K).

Definition 27 (Category Hit Ratio (CHR@K))**.** *Let $\mathcal{C}$ be the set of the classes extracted from the IFE, and $\mathcal{I}_c = \{i \in \mathcal{I}, c \in \mathcal{C} | F_c(x_i) = c\}$ be the set of items whose images are classified by the IFE in the c-class, e.g., the category of low recommended items. Then,*

we define categorical hit (chit) as:

$$chit(u,k) = \begin{cases} 1, & \textit{if k-th item in the top-K} \in \mathcal{I}_c \\ 0, & \textit{if k-th item in the top-K} \notin \mathcal{I}_c \end{cases} \tag{5.1}$$

where categorical hit (chit(u,k)) is a 0/1*-valued function that is* 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*. Consequently, we define the* CHR@K *as follows:*

$$\mathrm{CHR_u@K} = \frac{1}{K}\sum_{k=1}^{K} chit(u,k) \tag{5.2}$$

Since CHR@K does not pay attention to the ranking of the adversarially attacked recommended items, we propose a novel rank-wise positional metric, named Category normalized Discounted Cumulative Gain, that assigns a *gain* to each considered ranking position. By considering a relevance threshold τ, we assume that each item $i \in \mathcal{I}_c$ has an ideal relevance value of:

$$idealrel(i) = 2^{(s_{max}-\tau+1)} - 1 \tag{5.3}$$

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

$$rel(k) = \begin{cases} 2^{(s_{ui}-\tau+1)} - 1, & \text{if } \textit{k-th item} \in \mathcal{I}_c \\ 0, & \text{if } \textit{k-th item} \notin \mathcal{I}_c \end{cases} \tag{5.4}$$

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 position in the result list. Analogously, we define Category Discounted Cumulative Gain (CDCG) as:

$$\mathrm{CDCG_u@K} = \sum_{k=1}^{K} \frac{rel(k)}{\log_2(1+k)} \tag{5.5}$$

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, we define the Ideal

Category Discounted Cumulative Gain (ICDCG@K) as follows:

$$\text{ICDCG@K} = \sum_{k=1}^{min(K,|\mathcal{I}_c|)} \frac{rel(k)}{\log_2(1+k)} \tag{5.6}$$

In practical terms, ICDCG@K indicates the score obtained by an ideal recommendation list that contains only relevant items.

Definition 28 (normalized Category Discounted Cumulative Gain)**.** *Let $\mathcal{C}$ be the set of the classes extracted from the IFE, $\mathcal{I}_c = \{i \in \mathcal{I}, c \in \mathcal{C} | F_c(x_i) = c\}$ be the set of items whose images are classified by the IFE in the c-class, i.e., the category of low recommended items. Let $rel(k)$ be a function computing the relevance of the k-th item of the top-K recommendation list, and* ICDCG@K *be the* CDCG *for an ideal recommendation list only composed of relevant items. We define the normalized Category Discounted Cumulative Gain (nCDCG), as:*

$$\text{nCDCG}_\text{u}\text{@K} = \frac{1}{\text{ICDCG@K}} \sum_{k=1}^{K} \frac{rel(k)}{\log_2(1+k)} \tag{5.7}$$

The nCDCG@K is ranged in an $[0,1]$ interval, where values close to 1 mean that the attacked items are recommended in higher positions, e.g., the attack is effective. In Information Retrieval, a logarithm with a base 2 is commonly adopted to ensure that all the recommendation list positions are discounted.

5.3 Experiments

Here, we present experimental settings and the discussion of the empirical results.

5.3.1 Settings

In this section, we first introduce the three real-world datasets, the adversarial attack strategies, the defense methods to make the IFE more robust, and the VRSs. Then, we present the complete set of evaluation measures and a detailed presentation of the experimental choices to make the results reproducible.

Datasets

We experiment our models on the following datasets:

Table 5.1 Dataset statistics.

Data	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{R}\rvert$	density
`Amazon Men`	24,379	7,371	89,020	0.000495
`Amazon Women`	16,668	2,981	54,473	0.001096
`Tradesy`	6,253	1,670	21,533	0.002062

- `Amazon Men` and `Amazon Women` [159, 113, 114] are two datasets about men's and women's clothing from the Amazon category "Clothing, Shoes and Jewelry". Once having downloaded the images with a valid URL, we applied k-core filtering first on users and then on items to reduce the impact of cold users and items, as suggested by Rendle et al. [189]. While for `Amazon Men` we run 5-core filtering as suggested in [113, 114], for `Amazon Women` we adopted 10-core filtering to reduce its higher number of user/item interactions, and so reducing the VRS training time and the expensive hardware computation time in crafting adversarially perturbed product images [237]. This pre-processing step produced the following statistics: `Amazon Women` counts 54,473 interactions recorded between 16,668 users and 2,981 items, while `Amazon Men` count 89,020 interactions recorded 24,379 users and 7,371 items.

- `Tradesy` [114] dataset contains implicit feedback, i.e., purchase histories and desired products, from the homonym second-hand selling platform. We applied the same pre-processing pipeline described above. As for `Amazon Women`, we run *10*-core filtering. The final dataset counts 21,533 feedback recorded on 6,253 users and 1,670 products.

We report additional dataset statistics in Table 5.1.

Attacks

We test three state-of-the-art adversarial attacks against DNNs image classifiers.

- **Fast Gradient Sign Method (FGSM)** [101] is an L_∞-norm optimized attack that produces an adversarial version of a given image in just one evaluation step. A perturbation budget ϵ is set to modify the strength —and consequently, the visual imperceptibility— of the attack, i.e., higher ϵ values mean stronger attacks but also more evident visual artifacts.

- **Projected Gradient Descent (PGD)** [155] is a L_∞-norm optimized attack that takes a uniform random noise as the initial perturbation, and *iteratively* applies

an FGSM attack with a continuously updated small perturbation α —clipped within the ϵ-ball— until either it effectively reaches the network misclassification, i.e., $F_c(x_i + \alpha_i) = t$, or it completes the number of possible iterations, i.e., 10 iterations in our evaluation setting.

- **Carlini and Wagner attacks (C&W)** [57] are three attack strategies based on L_0, L_2 and L_∞ norms that re-formulate the traditional adversarial attack problem by replacing the distance metric with a well-chosen *objective function*. C&W integrates the parameters κ, i.e., the *confidence* of the attacked image being classified as t, and a, i.e., the trade-off between optimizing the objective function and the classifier loss function.

Defenses

We investigate two defense strategies.

- **Adversarial Training (AT)** [101] consists of injecting adversarial samples into the training set to make the trained model robust to them. The major limitations of this idea are that it increases the computational time to complete the training phase, and it is deeply dependent on the type of attack strategy used to craft adversarial samples. For instance, Madry *et al.* [155] generates adversarial images with the PGD-method to make the trained model robust against both one-step and multi-step attack strategies.

- **Free Adversarial Training (FAT)** [197] proposes a training procedure $3-30$ times faster than the classical Adversarial Training [101, 155]. Unlike the previous one, this method updates both the model parameters and the adversarial perturbations doing a unique backward pass in which gradients are computed on the network loss. Moreover, to simulate a multi-step attack —which would make the trained network more robust— it keeps retraining on the same mini-batch for m times in a row.

Visual-based Recommenders

To evaluate `VAR` approach, we considered five VRSs. Table 5.2 presents an overview of the IFE components of the tested VRSs.

- **Factorization Machine (FM)** [186] is a recommender model proposed by Rendle [186] to estimate the user-item preference score with a factorization technique.

Table 5.2 Technical details of the state-of-the-art visual recommenders tested in the experimental section of this chapter. We indicate with FC, Fully-Connected, and with FM, Feature Maps.

VRS		**Image Feature Extractor**			
		Extraction Layer		Training	
Model	*Reference*	*FC*	*FM*	*Pretrained*	*End-to-End*
FM	Rendle [186]	✓		✓	
VBPR	He and McAuley [114]	✓		✓	
AMR	Tang et al. [209]	✓		✓	
ACF	Chen et al. [61]		✓	✓	
DVBPR	Kang et al. [134]	✓			✓

For a fair comparison with VBPR and AMR, we used BPR [188] loss function to optimize the personalized ranking. In this respect, we adopted LightFM [141] implementation integrating $\mathcal{R}$ with the extracted continuous features. It is worth noticing that, differently from the recommenders we will present later, this model is not specifically designed for visually aware recommendation tasks.

- **Visual Bayesian Personalized Ranking (VBPR)** [114] improves the MF preference predictor by adding a *visual* contribution to the traditional *collaborative* one. Given a user u and a non-interacted item i, the predicted preference score is $\hat{s}_{ui} = \mathbf{p}_u^T \mathbf{q}_i + \theta_u^T \theta_i + b_{ui}$, where $\theta_u \in \Theta^{|\mathcal{U}| \times h}$ and $\theta_i \in \Theta^{|\mathcal{I}| \times h}$ are the *visual* latent vectors of user u and item i respectively ($h << |\mathcal{U}|, |\mathcal{I}|$). The visual latent vector of item i is obtained as $\theta_i = \mathbf{E}\varphi_i$, where φ_i is the visual feature of image item i extracted from a pretrained AlexNet [140] and $\mathbf{E}$ is a matrix to project the visual feature into the same space as of θ_u. Furthermore, b_{ui} includes the sum of the overall offset, and the user, item and global visual bias.

- **Attentive Collaborative Filtering (ACF)** [61] tries to unveil the *implicitness* of multimedia user/item interactions by means of two *attention* networks. That is, one network learns to weight each user's interacted, i.e., positive items — because they are not equally *important* to the user— while another network learns to weight each *component* of the *feature map* extracted from the product image within the interacted items, e.g., regions of an image or frames of a video. Given a user u and a non-interacted item i, the predicted preference

score is $\hat{s}_{ui} = \left(\mathbf{p}_u + \mathbf{v}_u\right)^T \mathbf{q}_i$, where $\mathbf{v}_u \in \mathbf{V}^{|\mathcal{U}| \times h}$ is an additional *user latent* vector weighted by the two *attention*-levels, i.e., item and component, described above.

- **Visually-Aware Deep BPR (DVBPR)** [134] enhances the preference predictor proposed by He and McAuley [114] by replacing the pretrained visual feature extractor with a custom Convolutional Neural Network (CNN), which is trained *end-to-end* together with the preference predictor on the main recommendation task. Given a user u and a non-interacted item i, the predicted preference score is $\hat{s}_{ui} = \theta_u^T F^{(e)}(x_i)$, where θ_u is the user visual profile seen for VBPR and F is the custom CNN.

- **Adversarial Multimedia Recommendation (AMR)** [209] is an extension of VBPR that integrates the adversarial training procedure proposed by He *et al.* [115] named *adversarial regularization* to build a model that is increasingly robust to FGSM-based perturbations against image features. Apart from the different training procedures, the score prediction function is the same as VBPR.

Evaluation Metrics

In addition to CHR@K and nCDCG@K, we also study both the effects of adversarial images on the IFE and the variation caused on the global recommendation performance.

IFE Performance. IFE performance is evaluated through the attack Success Rate (SR), the percentage of adversarial samples that have affected the classifier behavior, and the Feature Loss (FL), i.e., the mean squared error between the extracted image features before and after the attack, and the Learned Perceptual Image Patch Similarity (LPIPS) [241]. The idea behind LPIPS is to produce a perceptual distance value between two similar images by leveraging (1) knowledge extracted from convolutional layers inside state-of-the-art CNNs and (2) collected human visual judgments about those pairs of similar images. We computed this metric fine-tuning a VGG [203] network since Zhang et al. [241] proposed this configuration as the best one at imitating a real human evaluation in the circumstances comparable to visual attacks.

VRS Performance. Global recommendation performance is evaluated with Re@K, shown in Definition 4, and the expected free discovery (EFD@K), a beyond-accuracy metric that provides a measure of the ability of an RS to recommend relevant long-tail items [213]. Since we are interested in measuring whether the application of targeted adversarial attacks might alter the overall performance of the RS, Table 5.7 reports

the percentage variation of the performance between the attacked recommender and the base one. The reported metric is evaluated as follows:

$$\Delta_{\text{Rec}} = \frac{\frac{1}{|Attacks|}\left(\sum_{a \in Attacks} \text{Rec}_a\right) - \text{Rec}_{Base}}{\text{Rec}_{Base}} \times 100 \tag{5.8}$$

where *Attacks* indicates the set of tested attacks, e.g., FGSM, PGD, and C&W, and *Base* indicates that the metric value has been computed on the not-attacked recommender. The same formulation has been used to evaluate the Δ_{EFD}. Note that Δ negative values indicate a reduction of the performance.

Evaluation Protocol

Here, we present the evaluation strategies used in the experimental phase to reproduce our results.

Adversarial Attacks. We use the Python library CleverHans [177] to implement the attacks. For both FGSM and PGD, we adopt $\epsilon = 4$ re-scaled by 255. Then, for PGD's α parameter, we set the multi-step size as $\epsilon/6$ and the number of iterations to 10. As for the C&W attack, we run a 5-step binary search to calculate a, starting from an initial value of 10^{-2} and set κ to 0. Furthermore, we set the maximum number of iteration to 1000 and adopted Adam optimizer with a learning rate of 5×10^{-3} as suggested in C&W [57]. Finally, we save the adversarial images in `tiff` format, i.e., a lossless compression, as lossy compression, e.g., JPEG, may affect the attacks' effectiveness [106].

Image Feature Extraction. Image features extracted using the PyTorch pretrained implementation of ResNet50 [112]. For FM, VBPR, and AMR, we set `AdaptiveAvgPool2d` as extraction layer, whose output is a 2048-dimensional vector. For ACF, we set the last `Bottleneck` output, i.e., its final `relu` activation, as extraction layer, whose output is a $7 \times 7 \times 2048$-dimensional vector. Finally, for DVBPR, we reproduce the exact same CNN architecture described in the original paper [134], whose extraction layer output is a 100-dimensional vector. Here, we adopted TensorFlow.

Defenses. In the non-defended scenario, we adopt ResNet50 pre-trained on `ImageNet` with traditional training. On the other hand, we adopt ResNet50 pre-trained on `ImageNet` with Adversarial Training and Free Adversarial Training when applying defense techniques. For the former, we use a model trained with $\epsilon = 4$. For the latter,

Table 5.3 Averaged origin-target CHR on defence-free settings.

Dataset	***Origin***	CHR@K	***Target***	CHR@K	$\text{CHR}_\text{T}/\text{CHR}_\text{O}$
`Amazon Men`	Sandal	0.4508	Running Shoe	2.0191	4.4787
`Amazon Women`	Jersey, T-shirt	0.6324	Brassiere, Bandeau	1.8531	2.9305
`Tradesy`	Suit	0.3810	Trench Coat	1.5371	4.0345

Algorithm 2 Experimental Scenario of `VAR`.

1: Train the VRS on clean item images.
2: Measure the *Base* CHR@K for each category $\mathcal{C}$.
3: Select origin (O) and target (T) categories s.t. CHR_O@K $<$ CHR_T@K.
4: Perform an Adv. Attack against IFE to misclassify O-Images as T.
5: Poison the dataset with the adversarial perturbed item images.
6: Measure the HR_O@K of the O-Products after the Adv. Attack.

we employ a model trained with $\epsilon = 4$ and $m = 4$ (that explains why we only run attacks with $\epsilon = 4$). Both models are available in the published repository.

Recommenders. We realize FM using the LightFM library [141] training the model for 100 epochs and left all the parameters with the library default values. All the other models are implemented in TensorFlow. As for VBPR and AMR, we train the models following the training settings adopted by Tang et al. [209] while for DVBPR, we adopted the same parameters found in the official implementation [1]. On the contrary, we chose ACF hyper-parameters through *grid search* (batch size: $[32, 64, 128]$, learning rate: $[0.01, 0.1]$, regularizer: [0, 0.01, 0.001]). Learning rate and regularizer are set to 0.1 and 0 respectively, while the batch size is set to 32 for `Tradesy` and 64 for `Amazon Women` and `Amazon Men`. The rationale behind the fact that we apply a grid-search to test ACF is that the other VRSs are originally presented and trained in a highly comparable scenario to ours, i.e., the same datasets, while ACF has been tested by Chen et al. [61] on diverse datasets. For each dataset, we use the *leave-one-out* training-test protocol, putting in the test set the last time-aware user's interaction.

5.3.2 Results and Discussion

The research questions that will be addressed in this section are defined as follows:

RQ1 Which are the effects of targeted adversarial attacks on the IFE used in the VRSs in both defense-free and defense-activated settings?

[1] https://github.com/kang205/DVBPR

RQ2 Starting from the performance of the CNN used for the IFE, which are the effects of adversarial attacks and defenses on the VRS?

RQ3 How much the performance mentioned above is stable when we increase the length K of recommendation lists?

RQ4 Are the global recommendation performance worsened in the studied adversarial settings?

We present and discuss the `VAR` results evaluated on top-20 recommendation lists (we indicate $CHR@20$ as CHR). In this section, we adopt the notation <dataset, VRS, attack, defense> to indicate a specific `VAR` experimental setting. The reported results have been computed following Algorithm 2. Table 5.3 shows the statistics of the categories used in `VAR` experiments.

Attacks and Defenses Performance of IFE (RQ1)

This paragraph analyses the success rate (SR) and the feature loss (FL) of the adversarial attacks against the IFE components reported in Table 5.4. Since we did not apply any defensive strategy to the custom DNN adopted for DVBPR, the corresponding table cells have been left blank.

Attack Success Rate. Table 5.4 confirms PGD and C&W as the strongest attacks when applied to reduce the classification accuracy of a defense-free CNN classifier. For instance, PGD reaches a near-100% SR on `Amazon Men` and 100% SR on `Tradesy`, C&W's SR is always more than 89%, while FGSM never gets the same results, showing the lowest performance, i.e., 18%, on `Amazon Women`. As expected, this behavior varies with defense strategies. Under this setting, C&W emerges as the best offensive solution against defense strategies, as already demonstrated in [57]. For example, we observe an average SR reduction in the SR results of 77% for FGSM, 82% for PGD, and 62% for $C\&W$.

Hence, we compare the SR results to the variation of visual-aware recommendations for the items belonging to the perturbed category of images. Our assumption here is to empirically find a *conformity* between *classification* and *recommendation* metrics on the definition of *successful* attack. Surprisingly, Table 5.6 shows a different trend from the one observed earlier for the defense-free setting. As far as the CHR is concerned, FGSM and C&W attacks are almost aligned on average, i.e., 0.6222 and 0.6212 respectively, but PGD is the best performing attack, i.e., 0.7932 averagely. We also see *discrepancies*

Table 5.4 Average values of Success Rate (SR) and Feature Loss (FL) for each combination. FL values are multiplied by 10^3.

Data	VRS	Att.	Image Feature Extractor					
			Traditional		Adv. Train.		Free Adv. Train.	
			SR	FL	SR	FL	SR	FL
Amazon Men	FM, VBPR, AMR	FGSM	65%	14.0948	18%	0.0330	15%	0.0278
		PGD	**97%**	**36.8843**	18%	0.0334	15%	0.0283
		C&W	89%	20.5172	**48%**	**2.8022**	**42%**	**1.9080**
	ACF	FGSM	65%	9.0480	18%	0.0944	15%	0.0951
		PGD	**97%**	9.2606	18%	0.0944	15%	0.0954
		C&W	89%	**10.4917**	**48%**	**0.7582**	**42%**	**0.4955**
	DVBPR	FGSM	65%	**16.4055**	—	—	—	—
		PGD	**97%**	16.1151	—	—	—	—
		C&W	89%	16.3442	—	—	—	—
Amazon Women	FM, VBPR, AMR	FGSM	18%	9.6677	0%	0.0113	0%	0.0094
		PGD	85%	**27.6645**	0%	0.0119	0%	0.0102
		C&W	**89%**	21.2380	**6%**	**0.1770**	**6%**	**0.3376**
	ACF	FGSM	18%	9.3257	0%	0.0346	0%	0.0424
		PGD	85%	8.3596	0%	0.0352	0%	0.0436
		C&W	**89%**	**11.2079**	**6%**	**0.0399**	**6%**	**0.0594**
	DVBPR	FGSM	18%	20.6968	—	—	—	—
		PGD	85%	17.2065	—	—	—	—
		C&W	**89%**	**24.4750**	—	—	—	—
Tradesy	FM, VBPR, AMR	FGSM	83%	21.4011	43%	0.0308	30%	0.0274
		PGD	**100%**	**53.4589**	43%	0.0311	30%	0.0273
		C&W	**100%**	25.9374	**80%**	**2.1185**	**63%**	**1.9739**
	ACF	FGSM	83%	14.6235	43%	0.0912	30%	0.1069
		PGD	**100%**	10.7754	43%	0.0899	30%	0.1044
		C&W	**100%**	**15.6256**	**80%**	**1.8834**	**63%**	**1.5343**
	DVBPR	FGSM	83%	24.7173	—	—	—	—
		PGD	**100%**	27.0801	—	—	—	—
		C&W	**100%**	**33.6879**	—	—	—	—

under defense-activated scenarios, in which all calculated CHR values show negligible differences, with FGSM and C&W mildly outperforming PGD, i.e., especially on AT.

Observation 1. Attack success rate is not directly related to the effects on the recommendation performance. In other words, being powerful enough to lead a classifier in mislabelling an origin product image towards a target class does not justify the recommendation lists' effects.

Features Loss. Motivated by the previous observations, we investigate the Feature Loss (FL) between original and attacked samples (as shown in Table 5.4). The "VRS" column combines the models according to both the IFE and the extraction layer used in the recommendation task. Our assumption here is to empirically find that high

Table 5.5 Average values of Learned Perceptual Image Patch Similarity (LPIPS) for Amazon Datasets combination. LPIPS is multiplied by 100. We mark in **bold** the best results.

Data	**Attack**	**Image Feature Extractor**		
		T	**AT**	**FAT**
		LPIPS	LPIPS	LPIPS
`Amazon Women`	FGSM ($\epsilon = 8$)	2.8505	1.8298	1.2119
	PGD ($\epsilon = 8$)	1.1136	0.7683	0.6369
	C & W	0.2678	**0.0731**	**0.0816**
`Amazon Men`	FGSM ($\epsilon = 8$)	1.7124	2.2903	1.2293
	PGD ($\epsilon = 8$)	0.6916	0.7997	0.6468
	C & W	0.2279	0.2688	0.1490

distances in the *feature* space correspond to high values of CHR and nCDCG (we leave the SR out of the discussion due to the previous finding). Comparing the results in Tables 5.4 and 5.6, we confirm a correlation between the variation of FL and the attack efficacy on VRSs. For instance, we see how PGD and C&W higher adversarial power in poisoning the VRS on `Amazon Women`—both on traditional and defensive scenarios— is also evident in the calculated FL on the same dataset. Additionally, we notice that the FL obtained for DVBPR on `Amazon Women` and `Tradesy` is averagely higher than the one on `Amazon Men`, i.e., 20.7928 and 28.4951 on `Amazon Women` and `Tradesy` respectively *vs.* 16.2883 on `Amazon Men`. We also identify the same trend on DVBPR from a *recommendation* point of view, i.e., there could be an attack method able to increase the *base*-case CHR.

Observation 2. The modification of VRS is closely linked to the magnitude difference between original and perturbed image features. In short, perturbations leading to more significant feature modifications may cause a strong influence on the recommendability of the altered items.

LPIPS. Table 5.5 reports the LPIPS values measured on the Amazon datasets. We observe that all attack combinations can keep LPIPS values within low ranges, under the *imperceptible* nature of adversarial perturbations on images [208]. Thus, we connect this obtained measure with the attack efficacy in failing the classifier (i.e., the DNN) and the VRS. What follows is a detailed evaluation of scenarios involving —or not— defensive techniques for the DNN. FGSM ($\epsilon = 8$) fails to hide the produced perturbations in the defense-free scenario, reaching the highest perceptible visual difference on `Amazon Women` (2.8505). Coherently, this setting also shows a low SR and a weak alteration

of visual recommendations (see Tables 5.4 and 5.6). Focusing on the two defenses becomes fundamental to consider the LPIPS value along with its corresponding SR and recommendation variations. As a matter of fact, in a defense context, where all attacks averagely tend to perform worse at failing the DNN classifier, a measured low average LPIPS value might trivially mean *very few* images were *successfully* attacked. For instance, the described situation occurs in the combination <`Amazon Men`, PGD ($\epsilon = 8$), AT>. However, since these attacks have still been effective in *pushing* low ranked category products (as evident in Table 5.6), then adversaries could exploit their imperceptibility to craft even stronger perturbations (e.g., increasing ϵ). An intriguing situation is when LPIPS on the defended DNN is higher than the non-defended one. The worst case is <`Amazon Men`, FGSM ($\epsilon = 8$), AT>, which shows a 34% increase of LPIPS compared to the Traditional training. We explain this result by considering that an attack might need to produce more significant perturbations to move the category of the few correctly attacked images (about 24% in the cited example) towards the targeted one. Not only is the attack inefficient, but it risks human identification.

Observation 3. The offline analysis of the possible human imperceptibility of adversarial perturbations with the state-of-the-art metric LPIPS have demonstrated that attacked images have barely perceptible visual artifacts that still keep breaking recommendation performance are blind spots that adversaries could explore deeper for their malicious purposes.

Category-based Performance (RQ2)

After having justified the results in Table 5.6, we discuss the category-based measures across models and datasets studying the CHR and nCDCG.

The results on FM show that adversarial attacks are always effective in the case of defense-free settings, with an across-dataset average CHR and nCDCG improvements of +5.46% and 6.51%, respectively. Furthermore, the application of the two defenses shows a partial defense. For instance, the <`Amazon Men`, FM, (AT, FAT)> combinations verify that the recommendability of the perturbed category could even receive small negative variations, e.g., an average reduction of CHR of -5.94% in the AT case. However, it can be seen that attacks are still effective in any <(`Amazon Women`, `Tradesy`), AT, FM> scenarios, e.g., $CHR_{PGD} = 0.4854 > CHR_{Base} = 0.4720$ in the `Amazon Women` dataset.

As regards VBPR, PGD is the most impactful strategy in any defense-free setting. For instance, PGD leads to a three times CHR increase of the attacked category, i.e., suit, on the `Tradesy` dataset. It means that the adversary has been able to push the class of products in the recommendation lists very effectively, ensuring that a

Table 5.6 Results of the `VAR` framework. A CHR@K, or nCDCG@K, higher than the *Base* means that the attack is effective. For each <dataset, VRS, defence> combination we put in bold the most efficient attack.

Data	VRS	Att.	Image Feature Extractor					
			Traditional		Adv. Train.		Free Adv. Train.	
			CHR	nCDCG	CHR	nCDCG	CHR	nCDCG
Amazon Men	FM	*Base*	*0.4960*	*0.0246*	*0.4082*	*0.0204*	*0.4048*	*0.0202*
		FGSM	**0.5309** *	**0.0266***	0.3886	0.0198*	0.3821*	0.0194*
		PGD	0.5293*	**0.0266***	0.3795*	0.0193*	0.3811*	0.0193*
		C&W	0.5258*	0.0263*	0.3837*	0.0194*	0.3871*	0.0194*
	VBPR	*Base*	*0.6531*	*0.0293*	*0.3074*	*0.0141*	*0.3775*	*0.0159*
		FGSM	0.5824*	0.0299	0.6164*	0.0323*	0.5860*	0.0283*
		PGD	**1.1480**	**0.0538***	0.6410*	0.0324*	0.5918*	0.0286*
		C&W	0.6132*	0.0290	**0.6880***	**0.0336***	**0.6642***	**0.0348***
	AMR	*Base*	*0.3944*	*0.0196*	*0.5037*	*0.0232*	*0.1076*	*0.0038*
		FGSM	0.3347*	0.0150*	0.4426*	0.0235	0.4178*	0.0187*
		PGD	**0.8365**	**0.0418***	0.4519*	**0.0242**	0.4263*	0.0193*
		C&W	0.3678	0.0170*	0.4371*	0.0230	**0.4451***	**0.0202***
	ACF	*Base*	*0.5574*	*0.0278*	*0.3560*	*0.0176*	*0.3565*	*0.0176*
		FGSM	**0.5692***	**0.0282***	**0.3773***	**0.0185***	0.3517	0.0172*
		PGD	0.5610	0.0280	0.3731*	0.0183*	0.3521	0.0172*
		C&W	0.5628	0.0279	0.3690*	0.0181*	0.3471*	0.0169*
	DVBPR	*Base*	*0.6945*	*0.0359*	—	—	—	—
		FGSM	0.6579*	0.0329*	—	—	—	—
		PGD	0.5549*	0.0281*	—	—	—	—
		C&W	0.6414*	0.0306*	—	—	—	—
Amazon Women	FM	*Base*	*0.6956*	*0.0347*	*0.4720*	*0.0236*	*0.3231*	*0.0162*
		FGSM	0.7030	0.0354*	0.4804*	0.0243*	0.3022*	0.0150*
		PGD	**0.7144**	**0.0356***	**0.4854***	**0.0244***	0.3093*	0.0155*
		C&W	0.6935	0.0346	0.4761*	0.0240	0.2877*	0.0144*
	VBPR	*Base*	*0.4475*	*0.0210*	*0.5213*	*0.0251*	*0.3476*	*0.0161*
		FGSM	0.3933*	0.0182*	0.6199*	0.0310*	0.6204*	0.0318*
		PGD	**0.9530***	**0.0459***	**0.6463***	**0.0327***	**0.6413***	**0.0330***
		C&W	0.4215*	0.0179*	0.6457*	0.0326*	0.5880*	0.0302*
	AMR	*Base*	*0.9907*	*0.0462*	*0.8640*	*0.0454*	*0.5207*	*0.0303*
		FGSM	**1.4178***	**0.0862***	0.7379*	0.0334*	0.4658*	0.0230*
		PGD	1.2720*	0.0713*	0.6664*	0.0307*	0.5003*	0.0250*
		C&W	1.3762*	0.0761*	0.7390*	0.0336*	0.5112*	0.0252*
	ACF	*Base*	*0.9903*	*0.0511*	*0.6890*	*0.0349*	*0.4338*	*0.0219*
		FGSM	0.9895	0.0509	0.6935	0.0350	0.4737*	0.0242*
		PGD	0.9932	0.0512	0.6915	0.0348	0.4759*	**0.0243***
		C&W	**0.9947**	**0.0514***	**0.6943**	**0.0351**	**0.4774***	**0.0243***
	DVBPR	*Base*	*0.7787*	*0.0370*	—	—	—	—
		FGSM	0.7959*	0.0388*	—	—	—	—
		PGD	0.7407	0.0385*	—	—	—	—
		C&W	**0.9002***	**0.0436***	—	—	—	—
Tradesy	FM	*Base*	*0.3424*	*0.0167*	*0.3629*	*0.0183*	*0.4774*	*0.0241*
		FGSM	0.3696*	0.0183*	0.3800*	0.0189	0.5234*	0.0268*
		PGD	0.3664*	0.0180*	0.3661*	0.0181	0.5172*	0.0265*
		C&W	**0.3800***	**0.0190***	**0.3968***	**0.0196***	**0.5236***	**0.0269***
	VBPR	*Base*	*0.4201*	*0.0213*	*0.3011*	*0.0139*	*0.3243*	*0.0146*
		FGSM	0.5313*	0.0293*	**0.5182***	**0.0277***	**0.5770***	**0.0294***
		PGD	**1.3126***	**0.0748***	0.4508*	0.0226*	0.5330*	0.0268*
		C&W	0.4603*	0.0251*	0.4884*	0.0252*	0.5612*	0.0274*
	AMR	*Base*	*0.3710*	*0.0174*	*0.1638*	*0.0065*	*0.2215*	*0.0094*
		FGSM	0.4855	0.0246*	**0.3662***	0.0190*	**0.4094**	**0.0200***
		PGD	**1.0768***	**0.0585***	0.3490*	0.0180*	0.3683*	0.0181*
		C&W	0.4372*	0.0214*	0.3648*	**0.0196***	0.3672*	0.0172*
	ACF	*Base*	*0.3712*	*0.0192*	*0.3685*	*0.0178*	*0.4476*	*0.0218*
		FGSM	**0.3774***	**0.0195***	0.3864*	0.0189*	**0.4606***	**0.0223**
		PGD	0.3728	0.0193	0.3869*	**0.0190***	0.4604*	**0.0223**
		C&W	0.3734	0.0193	**0.3875***	**0.0190***	0.4561*	0.0221
	DVBPR	*Base*	*0.5810*	*0.0298*	—	—	—	—
		FGSM	**0.5956***	**0.0365***	—	—	—	—
		PGD	0.4668*	0.0238*	—	—	—	—
		C&W	0.5701*	0.0308*	—	—	—	—

** denotes statistically significant results (p-value* ≤ 0.05*).*

suit will be recommended at least one time for each top-20 recommendation list, i.e., $CHR = 1.3126 > 1$ in the <`Tradesy`, VBPR, PGD, T> setting. Additionally, we observe that there are effective attacks in any defended setting.

Observation 4. The adversarial robustification strategies have not protected VBPR from the injection of perturbed images, although they got high performance in protecting the classification.

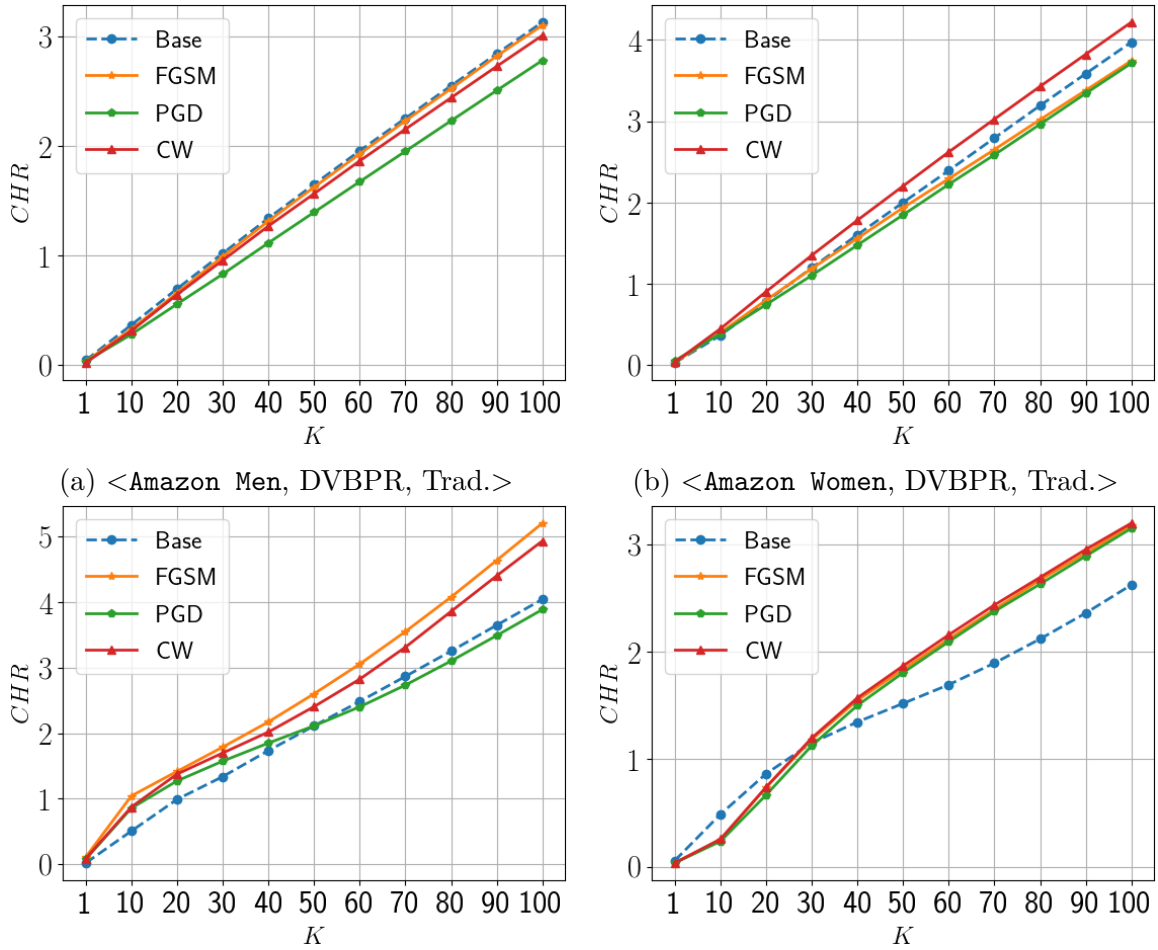
The third tested VRS is AMR. We chose this model since it is the first VRS to **integrate adversarial protection by design**, so we expected to get a limited variation in traditional performance under attack settings. Surprisingly, results show that AMR is prone to the effects of attacks as much as VBPR. For example, the PGD method represents the biggest security threat on the VRS in defense-free settings, with an average CHR improvements of +48.84% across the three datasets. Moreover, we observe that <AMR, (AT, FAT)> models do not protect the proposed adversarial threat model, notwithstanding the two defense techniques applied on both the IFE and the VRS, respectively. For instance, $CHR = 0.4451 > 0.1076$ when comparing *C&W* and *Base* in <`Amazon Men`, AMR, FAT> experiments. We justify AMR's *low-quality* protection against the tested attacks by the fact that it applies the adversarial regularization directly on the extracted visual features [209], whereas in our experimental framework, the perturbation is produced at the pixel level.

Observation 5. Combining state-of-the-art adversarial robustification of the IFE, e.g., AT and FAT, and the adversarial robustification of the VRS, e.g., the adversarial regularization of an RS [115]) does not guarantee the protection of the performance.

The fourth model is ACF. This model is the most robust in the case of defense-free settings when compared with the other models that use the visual features extracted from an external pre-trained IFE, i.e., FM, VBPR, and AMR. Indeed, both CHR and nCDCG show average variations of +0.79% and 0.61%, respectively, that are much smaller than the one observed in the other models, e.g., the variation is 44.71% in VBPR experiments. The same limited adversary efficacy in altering the recommendation lists can also be seen in the defended settings.

Observation 6. The tendency of ACF to be naturally robust to the tested attacks can be associated with the fact that it integrates a more semantic-oriented latent representation of the images, e.g., the feature map, and its recommendation task depends not only on the features extracted from the attacked item but also from the set of the items previously voted by each user.

Finally, we study whether the attacks against a pre-trained CNN used for image classification are transferable to DVBPR, a VRS that learns the deep visual features



(a) <`Amazon Men`, DVBPR, Trad.>

(b) <`Amazon Women`, DVBPR, Trad.>

(c) <`Amazon Women`, AMR, Trad.>

(d) <`Amazon Women`, AMR, Adv. Train.>

Fig. 5.2 Plots of $CHR@K$ by varying K from 1 to 100 on DVBPR and AMR trained on `Amazon Men` and `Amazon Women`.

within the downstream recommendation task. It can be seen that the adversary's efficacy depends on the attacked dataset. Indeed, results in Table 5.6 show that DVBPR is not affected by an increase of CHR in the `Amazon Men` dataset. However, we can see that C&W effectively varies CHR by more than the +10% in the `Amazon Women` dataset, and FGSM changes the CHR by +2.52% in `Tradesy`.

Observation 7. The learning of personalized deep visual representation of product images by DVBPR could be fooled by adversarial attacks transferred from another-trained DNN, raising the need for further investigation to robustify these models.

Attack results when increasing the length of top-K lists (RQ3)

Before we move to the study of overall recommendation performance, we investigate the effects of adversarial attacks and defenses by varying the length of recommendation lists (K). Figure 5.2 reports two plots related to possible interesting cases shown in Table 5.6: (1) the case where DVBPR was robust, or not, against the tested attacks, and (2) the case where, by changing the IFE from a traditional to an adversarial trained one, AMR showed more robust CHR@20 results in the `Amazon Women` dataset. The first scenario in Figure 5.2a shows that the robust behavior of DVBPR observed in the `Amazon Men` dataset is also confirmed on top-100 recommendation lists, while Figure 5.2b verifies that C&W sill is a powerful strategy to push the perturbed category of product with the difference with the CHR@K-baseline that increases with K. Regarding the second set of plots, Figure 5.2c confirms that FGSM and C&W make the adversarial regularization of the VRS ineffective since the CHR@K is always larger than *Base* as k increases, while Figure 5.2d returns a new unknown phenomenon related to the fact that the robustification of <AMR, AT>, observed on short recommendation lists, e.g., K=20 in Table 5.6, could be not confirmed on longer recommendation lists, e.g., K=100 ($\text{CHR@100}_{\text{C\&W}} \simeq 1.22 \times \text{CHR@100}_{\text{Base}}$).

Observation 8. Adversarial attacks' efficacy might be even more evident when analyzing longer top-K lists, raising the need for more powerful defensive strategies in cases where the model is robust on short-length recommendation lists.

Overall Recommendation Variations (RQ4)

Table 5.7 reports the variations of Re and EFD measured on attacked recommenders. The aim is to understand whether the application of defenses adopted to alleviate attacks' influence could generate a drastic variation of the overall recommendation performance. For instance, Δ_{EFD} on AMR has positive values independently of the application of defense mechanisms in the case of `Amazon Men`, i.e., $\Delta_{EFD} = +14.74\%$ in the case of FAT defense. In contrast, VBPR gets more negative variations across both metrics in the cases tested on the `Amazon Men` dataset. This behavioral pattern is different in the case of `Amazon Women`. Indeed, VBPR measures get positive variation for FAT experimental cases, e.g., $\Delta_{\text{Rec}} = +5.53\%$ on the Traditional model, while negative for the AT one, e.g., $\Delta_{\text{Rec}} = -10.51\%$.

Observation 9. The application of powerful attacks has not tragically worsened the accuracy and beyond accuracy performance. On the contrary, some measures have significantly improved as a consequence of the attack.

Table 5.7 Results of the overall variations of two recommendation metrics: recall (Rec) and expected free discovery (EFD) to understand whether the tested attacks can be identifiable by looking at the overall RS performance.

Data	VRS	Image Feature Extractor					
		Traditional		Adv. Train.		Free Adv. Train.	
		Δ_{Rec}	Δ_{EFD}	Δ_{Rec}	Δ_{EFD}	Δ_{Rec}	Δ_{EFD}
Amazon Men	FM	+8.00	+38.45	-30.08	-18.04	-4.52	-4.17
	VBPR	+2.37	-1.33	-45.49	-41.58	-31.42	-33.76
	AMR	+0.75	+1.37	+5.92	+14.74	+2.50	+9.97
	ACF	-1.54	-4.02	-0.69	+0.35	+6.19	0.00
	DVBPR	+6.17	+4.72	—	—	—	—
Amazon Women	FM	+8.42	+0.81	+23.69	+20.82	+9.02	+9.59
	VBPR	-1.74	-0.95	-10.51	-13.47	+1.29	+3.39
	AMR	-0.26	-1.39	+6.04	+5.71	+5.34	+3.90
	ACF	-1.96	-1.74	+1.72	-4.32	+5.50	+10.95
	DVBPR	-0.24	+2.94	—	—	—	—
Tradesy	FM	+5.23	-0.23	+8.51	+11.01	+36.59	+27.7
	VBPR	+2.95	-0.51	+4.50	-4.71	-1.17	-9.85
	AMR	+17.92	+20.88	+24.82	+28.98	+3.48	-2.38
	ACF	-2.38	-2.20	-6.17	-15.55	-4.95	-11.00
	DVBPR	-11.11	-15.47	—	—	—	—

Analyzing the overall variations across the VRS, we observe that ACF and DVBPR are the models less likely to get substantial overall performance variations when under attacks. For instance, ACF shows a total average variation of -1.22%, while DVBPR by -2.17%. On the contrary, FM, VBPR, and AMR are the models with less stable overall recommendations. For example, VBPR gets overall variations on both metrics higher than -11%, while AMR shows variations close to $+9\%$.

Observation 10. Both the ACF attentive mechanisms and the DVBPR personalized image features extracted make the recommendation task less subjected to performance variations when the images of a single category of products are perturbed towards a target (popular) one.

5.4 Related Work

The integration of image features in user's preference predictor leads to enhancing both recommendation [113, 114, 170, 238, 64] and search [228, 136, 238] tasks. The intuition

is that the visual appearance of product images influences customer's decisions, e.g., a customer who loves red will likely buy red clothes [102]. For instance, He and McAuley [114] extended BPR-MF [188] by integrating high-level features extracted from a pre-trained CNN, while Kang et al. [134] trained the same model in an end-to-end manner by stacking a custom CNN at the top, whose purpose is feature representation learning and not simply classification. Yu et al. [233] added aesthetic information in the recommendation framework to enhance CNNs' extracted features, which carry only semantic content. Yin et al. [232] proposed to incorporate visual features to learn item-to-item compatibility relations for outfit recommendation. Furthermore, Niu et al. [170] injected the visual features into a personalized neural model, and Chen et al. [61] integrated component-level image features, e.g., regions in an image, to learn users' preferences from more informative image representations. In this chapter, we have focused on VRSs that integrate both features extracted from both CNNs pre-trained for a *classification task*, e.g., [113, 61, 170, 209], and CNNs learned within the VRS [134] to tackle an adversary threat model whose goal is to push a category of products thanks to the capability of perturbing item images to be inserted in the dataset at training time. The adversarial works closest to the research topic explored in this chapter are the attack model proposed by Tang et al. [209] that applied adversarial perturbations on the image features instead of images, and the works by Cohen et al. [67], Liu and Larson [154] that have studied testing time pixel-level adversarial attacks. In contrast, our threat model explores training time.

5.5 Summary

We have presented an evaluation framework, i.e., Visual Adversarial Recommendation (`VAR`), to explore the application of targeted adversarial attacks (`BB-TAaMR`) on input images for multimedia recommenders and investigate the effectiveness of robustification mechanisms on the DNNs, i.e., Adversarial Training/Free Adversarial Training, used to robustify the image feature extractor of a visual recommender. We have tested three state-of-the-art white-box attacks, i.e., FGSM, PGD, and C&W, to perturb the images of low-recommended products with the adversaries' goal to make these pictures misclassified the DNN toward the class of top-rated products (and push their recommendability). Experimental results have shown that low recommended product categories could become up to three times more recommended by perturbing product images in a human-imperceptible way, and the defense mechanisms do not guarantee the protections of VRSs against attacks. Interestingly, we have found that

the effectiveness of attacks in altering the recommenders is more related to high feature losses than high success rates. Additionally, we have also observed that DVBPR, a VRS that learns deep image representations without external DNNs, is not robust to adversarial samples transferred by attacking other networks. Finally, we have verified that overall recommendation performance has not worsened under the experimented threat model and defended IFEs may even improve in non-attack settings. These findings raise the need to develop novel defense approaches to protect visually aware recommender models. Investigating the reasons behind the models' weakness could benefit the studied recommenders and verify whether other multimedia recommenders, e.g., music recommenders, could be affected by the same treats, e.g., push an artist.

Chapter 6

Adversarial Image Denoiser to Defend Multimedia RSs against Test-Time Attacks

Can Adversarial Image Denoiser (AiD) reduce the effectiveness of adversaries that use test-time adversarially-perturbed product images? How much AiD application is affecting the overall accuracy and beyond-accuracy performance?

Visual-based recommender systems (VRSs), have been demonstrated to be vulnerable to test-time adversarial examples— noised item images that are almost human-indistinguishable from clean ones— that, when integrated by a trained VRS, alter its reliability by recommending improper products. While stronger and stronger adversarial attacks have recently emerged to raise awareness of the risks, effective defense methods are still an urgent open challenge. Indeed, the state-of-the-art defensive strategy, named adversarial training for RSs, has been revealed to drastically fails under these malicious strategies. In this chapter, we propose "Adversarial Image Denoiser" (AiD), a novel defense method to protect VRSs against adversarial attacks. In AiD, we exploit the idea of cleaning up the product images by the perturbations added by the adversaries. In particular, we propose a U-Net-based denoising autoencoder trained to minimize the visual differences between clean and adversarial images while preserving the recommender systems' behavior in clean settings.

Compared with the adversarial trained VRS, AiD has three main advantages. First, it is easily integrable in existing visual recommendation methods (even with the adversarially trained ones) because it operates on the products' images before their use in the recommendation process. Second, the victim recommender protected by AiD

is more robust to either white- and black-box adversarial attacks by reducing their efficacy in changing the original model behavior on score prediction and top-K ranking tasks. Third, it preserves most of the overall recommendation performance measured in clean settings under accuracy and beyond-accuracy evaluation perspectives. Extensive experiments evaluate the efficacy of the proposed defense using three state-of-the-art adversarial attacks when mounted against standard visually-aware recommender algorithms on three real-world datasets.

6.1 Introduction

The economic gain associated with the use of RSs, together with the performance enhancement proved for their visually-aware variant, have made VRSs the target of adversaries [82]. For instance, an adversary can be an e-commerce competitor willing to boost her sales by uploading adversarially perturbed product images [209, 85, 67, 154, 18]. Tang et al. [209] are the first authors to propose adversarial attack procedures for reducing the accuracy of VRSs by altering the extracted image features with the perturbation method proposed by He et al. [115]. However, this chapter has assumed that the adversary should have edit access to the recommendation model parameters by making it impractical in a real-world scenario.

Subsequent works have focused on adversaries that perform their malicious goals (i.e., pushing an item or a set of items in high positions of the recommendation lists) by directly uploading adversarially perturbed product images. For instance, Di Noia et al. [85], Anelli et al. [18] proposed an adversarial attack strategy, named `BB-TAaMR`, that implements attack strategies designed against image classifiers (i.e., FGSM [101], PGD [155], and Carlini & Wagner [57]) to mislead the CNN used to extract the visual features in classifying the target items towards as port of a popular category of products. Note that `BB-TAaMR` is a black-box strategy since the adversaries access only items' popularity information publicly available on the platform. Further, Liu and Larson [154] and Cohen et al. [67] have proposed adversarial attacks that perturb product images to push an item by building perturbations by maximizing the preference scores predicted by the recommender. While they have proposed both white-box (WB) and black box (BB) strategies— by assuming different levels of adversaries' knowledge— we focus on the strongest (WB) ones. In particular, Liu and Larson [154] have built the Insider Attack (`WB-INSA`) perturbations by directly employing the gradients measured when maximizing the predicted preference score, and Cohen et al. [67], have used the Sign of the Gradient (`WB-SIGN`) to speed up the perturbation process.

Fig. 6.1 Overview of a Visual-based Recommender Systems protected by the Adversarial Image Denoiser (AiD) in the presence of an Adversarial Image (x^*).

While the literature on proposing novel adversarial attack strategies is rich, only a few works exist on finding solutions to defend visual recommenders against the existing solutions. To the best of our knowledge, Adversarial Multimedia Recommendation (AMR) [209] is the state-of-the-art defensive solution proposed in recommendation settings. In this model, Tang et al. [209] integrate VBPR with the adversarial personalized training procedure proposed by He et al. [115]. However, while AMR has been proved to be effective against the adversarial perturbations of the visual features [209], recent attacks by Di Noia et al. [85], Anelli et al. [18], Liu and Larson [154] have tested its limits against adversarial perturbation of product images with the goal to push their positions in the recommendation lists.

Motivated by the lack of adequate defense mechanisms, in this chapter, we proposed a novel defense mechanism named Adversarial Image Denoiser (AiD) to be integrated before the feature extraction process. The main idea of this defense is to learn how to remove the noise from the adversarial images constructed to alter the visually-aware recommendation task. Technically, we accomplish this by training a U-Net-based denoiser auto-encoder [149] with a high-level and recommendation-level guided loss function. The architectural schema of a VRS protected by AID is shown in Figure 6.1. To summarize, our main contributions are:

- the proposal of a novel defense solution, named Adversarial Image Denoiser (AiD), to protect VRSs against adversaries that can upload adversarial images on the recommendation platform to push the recommendability of target items;
- the study of the AiD robustness in comparison with AMR— the state-of-the-art adversarially trained recommender— by evaluating the variations of the predicted scores and the ranking-positions of the victim items with under three adversarial attack methods (i.e., `BB-TAaMR`, `WB-SIGN`, and `WB-INSA`) by also varying the number of pixels modifiable by the adversary (perturbation budget) and the number of iterations that the adversary can perform to build the malicious noise;

- the verification that the integration of AiD into a VRS does not drastically impact the overall accuracy and beyond-accuracy recommendation performance when used in genuine scenarios (without adversarial images);

We conduct experiments on three real-world datasets to validate the effectiveness of the proposed defensive solution.

6.2 Background and Related Work

In this section, we first describe some useful notations and present the used formalization. Due to the complexity of the scenario, in this chapter, we use a slight different formalization that is fully presented in this section. Let $\mathcal{U}$, $\mathcal{I}$, and $\mathcal{S}$ be the set of users, items, and score-based preference feedback, where $|\mathcal{U}|$, $|\mathcal{I}|$, and $|\mathcal{S}|$ indicate the size of each set, respectively. Then, $s_{ui} \in \mathcal{S}$ is valued when the user $u \in \mathcal{U}$ has previously interacted with the item $i \in \mathcal{I}$. For instance, in the case of implicit feedback, $s_{ui} = 1$ when u has purchased or clicked i (e.g., the product of an e-commerce catalog). We define the **item recommendation task** as the problem of producing a user's personalized list of un-interacted items with the goal to maximize her utility function. To build the recommendation list, a recommender system sorts by descending order the unseen items based on $\hat{s}_\Theta(\cdot)$, the preference score predicted by the recommender, where Θ are the learned model parameters.

6.2.1 Visual-based Recommender Systems

VRSs enhance the performance of the item recommendation task by exploiting item images. The intuition is that the visual appearance of product images influences customers' decisions, e.g., a client who prefers white shoes will likely purchase white dresses [102]. A standard approach is to integrate high-level representations of product images, also named visual features [125, 91, 223], extracted from a convolutional neural network (CNN). Let x_i denote the original image associated with the item $i \in \mathcal{I}$, and y_i the category. Let $f_\Phi : x \rightarrow y$ be a L-depth CNN to predict $p(y_i|x_i)$, let Φ its model parameters, and $f^l(x_i)$, $0 \leq l \leq L-1$ be the output of the l-th layer of f given the input x_i, then $\varphi_i = f^l(x_i)$ is a visual feature that can be used in a VRS. Then, a visual-based recommender model predicts the preference score for each (u,i)-pair by redefining the prediction function as follows:

$$\hat{s}_{ui} := \hat{s}_\Theta(f^l(x_i), u, i) = \hat{s}_\Theta(\varphi_i, u, i) \tag{6.1}$$

Commonly, l is the first fully connected layer after the last convolutional block [114, 170, 134]. This formalization is at the core of many visual recommenders. For instance, He and McAuley [114] propose the pivotal visual recommender, named VBPR, integrating into BPR-MF [188] the visual features extracted from the AlexNet [140] network trained on ImageNet [84]. Similarly, Yu et al. [233] also add aesthetic information to enhance the quality of the image representations. Furthermore, Yin et al. [232] incorporate visual features for outfit recommendation. Niu et al. [170] use a neural model to learn non-linear relations between visual features and users' preferences.

6.2.2 Existing methods for adversarial attacks

The first attack by Szegedy et al. [208] aim to force a machine learning (ML) model to have an incorrect behavior on perturbed images.

Definition 29 (Adversarial Attack)**.** *Given a classifier $f(\cdot)$, the **adversarial perturbation** δ of the adversarial sample $x^* = x + \delta$[1] x such that $f(x^*) \neq f(x)$ is defined as follows:*

$$\max_{\delta} \ \mathcal{L}_f(x+\delta,\, y), \quad s.t., \ \|\delta\|_\infty \leq \epsilon, \tag{6.2}$$

*where $\mathcal{L}$ is the network loss function and ϵ is the **perturbation budget**, typically chosen as small as possible such that the ∞-norm of the perturbation is below that limit to make it human-imperceptible.*

While numerous attacks have emerged in the computer vision domain (e.g., [101, 155, 57])), recently, adversarial issues have also emerged in the recommendation task. Three types of adversarial perturbations have raised the interest of the research community depending on the altered input [82]: the set of recorded feedback [143, 93], the model parameters [115, 209], and the content data [182, 67, 154]. The work presented in this chapter is placed in the last category whose standard adversarial goal is to crush the reliability of a trained VRS by pushing (or nuking) an item, or a set of items, towards a higher recommendation position or bigger predicted preference score. An intuitive example is a seller uploading maliciously perturbed images of her products to increase their probability of being frequently recommended by an e-commerce. Assuming no- or complete-knowledge of the recommender model, both black-box (`BB`) and white-box (`WB`) strategies have been designed to quantify the adversarial risks. Below, we describe the most representative techniques: `BB-TAaMR` [85], `WB-SIGN` [67], and `WB-INSA` [154].

[1] Note that we use x instead of x_i to indicate an image associated with an item i.

Targeted Adversarial Attack.

Di Noia et al. [85] propose to perturb item images such that the product class predicted by the CNN used to extract the features will misclassify them towards the category of more famous articles. The authors' intuition is to make the visual features extracted from the adversarial images of the victim items (e.g., white bag) closer to a "target" class of popular products (e.g., shoes). Between the set of the strategies in [85], we test PGD [142], the most influential one, defined below.

Definition 30 (`BB-TAaMR`)**.** *Given the clean image x, the popular class p, the number of steps T, and the budget ϵ, let $\alpha = 2.5 \cdot \epsilon/T$ be the perturbation size applied at each step, then PGD is defined as follows:*

$$x^{*,0} \leftarrow x \quad \text{// Genuine product image.} \tag{6.3}$$

$$x^{*,t} \leftarrow Clip_{x,\epsilon}\Big[x^{*,t-1} - \alpha \cdot \text{sign}(\nabla_{x^{*,t-1}} \mathcal{L}_f(x^{*,t-1}, p)\Big] \tag{6.4}$$

where $t \in \{1, 2, ..., T\}$, $\nabla_{x^{*,t}} \mathcal{L}_f(\cdot)$ is the Jacobian of $f(\cdot)$, $\text{sign}(\cdot)$ is the sign function, and $Clip_{x,\epsilon}\Big[\cdot\Big]$ is an element-wise clipping function to limit the L_∞-norm of the final perturbation in the ϵ-bound.

Sign-based Attack.

Cohen et al. [67] suggest to build a white-box attack by computing the sign of the gradient of the recommendation score function $\hat{s}(\cdot)$ with respect to all the pixels p_x in the product image x. In particular, the authors apply the chain rule to perform the gradient as follows: $\frac{\partial \hat{s}(x)}{\partial p_x} = \frac{\partial \hat{s}(x)}{\partial \varphi} \cdot \frac{\partial \varphi}{\partial p_x}$.

Definition 31 (`WB-SIGN`)**.** *Given x, p, T, ϵ, and α as in Definition 30, the `WB-SIGN` adversarial attack method is defined as follows:*

$$x^{*,t} \leftarrow Clip_{x,\epsilon}\Big[x^{*,t-1} + \alpha \cdot \text{sign}(\frac{\partial \hat{s}(x)}{\partial p_x})\Big] \tag{6.5}$$

where, to be fair in comparing with the other attacks, we have extended the initial formulation in [67] with an iterative implementation.

The authors also propose two `BB` strategies not explored in this chapter since, as expected, they have been demonstrated to be much less effective than `WB-SIGN`.

Insider Attack.

Liu and Larson [154] propose a `WB`-attack model, named insider attack (`WB-INSA`), by adding an adversarial perturbation on the item images through an iterative methodology to maximize the predicted scores over the users in the platform.

Definition 32 (`WB-INSA`)**.** *Given x, p, T, and ϵ as in Definitions 30 and 31, the* `WB-INSA` *adversarial sample is generated by*

$$x^{*,t} \leftarrow Clip_{x,\epsilon}\left[x^{*,t-1} + \frac{\partial \hat{s}(x)}{\partial p_x}\right] \tag{6.6}$$

where, to be comparable with the other strategies we preserve the maximum clipping of the perturbation in the ϵ bounded space.

`WB-INSA`, differently from `WB-SIGN`, directly uses the gradient back-propagated through the VRS and the CNN to alter the images.

6.2.3 Existing methods for defenses

Adversarial Training (AT) by Goodfellow et al. [101] is a popular defense strategy to robustify ML models by training them with adversarial samples. While its origins date back to the robustification of image classifiers, He et al. [115] have adapted the approach for the recommendation task by robustifying the recommender with respect to adversarial perturbations applied on model parameters.

Definition 33 (Adversarial Training for Recommenders)**.** *Let Θ and $\mathcal{L}_{RS}$ be the parameters and the loss of an RS, then the* ***Adversarial Training*** *for RSs is defined as follows:*

$$\arg\min_{\Theta} \max_{\delta_{adv}, \|\delta\| \leq \epsilon} \mathcal{L}_{RS}(\Theta) + \lambda \underbrace{\mathcal{L}_{RS}(\Theta + \delta)}_{\substack{adversarial \\ regularizer}} \tag{6.7}$$

$$with \quad \delta = \epsilon \cdot \frac{\Gamma}{\|\Gamma\|} \quad and \quad \Gamma = \frac{\partial \mathcal{L}_{RS}(\Theta + \delta)}{\partial \delta} \tag{6.8}$$

where λ is the adversarial regularization coefficient that controls the adversarial regularizer – the model loss obtained when an adversarial noise perturbs the model parameters.

Motivated by the AT efficacy in robustifying pure CF models [115, 235, 179, 71], Tang et al. [209] adapt this technique for designing robust multimedia recommendations. The proposed state-of-the-art defended VRS, named Adversarial Multimedia Recommender

(AMR), robustifies VBPR [114] by implementing the technique in Definition 33. However, AMR is trained to be robust against the visual features' perturbation (φ) and not the product images ones against whom it has been demonstrated to be fragile Anelli et al. [18], Liu and Larson [154]. For this reason, we propose AiD inspired by the intuition of removing the noise instead of making the recommender robust against it as effectively tested in other tasks such as image classification [103, 149].

6.3 The Proposed Defense

Since adversarial perturbations build with the attack strategies presented in Section 6.2 are constrained to be small at the pixel level, we propose the integration of an image denoiser to remove the adversarial noise and reduce the harmful effects of the attacks. Here, we describe the AiD architecture shown in Figure 6.2, followed by the presentation of its loss function and training algorithm.

6.3.1 Architecture

We implement AiD as a convolutional version of the denoising auto-encoder (DAE) [215] upgraded with a U-net [192], where $d_\Omega : x^* \rightarrow \widetilde{x}$ is the denoising function where Ω are the model parameters. The used architecture, named DUNET, has been designed by Liao et al. [149] to learn how to reconstruct the adversarial noise (dx) to be removed from the input adversarial sample such that:

$$\widetilde{x} = x^* - d\widetilde{x}. \tag{6.9}$$

where the denoised image $\widetilde{x}$ should be equal or similar to the clean one x and $d\widetilde{x}$, the AiD's learned adversarial noise, should be equal to δ (i.e., the adversarial perturbation added to x to make x^*). As shown in Figure 6.2, AiD is composed of a feedforward (encoder) and a feedback (decoder) path connected with lateral links (Fuse operation) going from the encoder layers to their corresponding decoder's one. Note that the input and output shapes are both 224x224x3 which are the input dimensions of the CNN used in our experiments (i.e., ResNet50 [112]).

6.3.2 Loss Function

A standard pixel-level guided denoiser (PixGD) loss is defined as $\mathcal{L}_{PixGD} = ||x - \widetilde{x}||$, where $||\cdot||$ is the L_1-norm. Liao et al. [149] have demonstrated that even if PixGD

suppresses the pixel-level noise, then the imperceptible adversarial perturbation can be progressively amplified by the network with the distortion of its high-level responses, which are the visual features (i.e., φ) used in a VRS (see Section 6.2.1). For this reason, we use a high-level guided denoiser (HGD) loss function.

Definition 34 (High-level Guided Loss)**.** *Let φ the item visual features of a clean image x, let $\widetilde{\varphi}$ the features extracted from its denoised version x^*, then the high-level guided denoiser (HGD) loss function is defined as follows:*

$$\mathcal{L}_{HGD} = ||\varphi - \widetilde{\varphi}|| \tag{6.10}$$

where the denoiser is explicitly trained to reconstruct the original visual feature (φ) lately used in the VRS.

Each product's visual feature produced by the denoised image (i.e., $\widetilde{\varphi}$) is then integrated in the recommender to infer the preference scores ($\hat{s}_\Theta(\widetilde{\varphi_i}, u, i)$). To make the training of the denoiser aware of preserving the preference scores predicted by the recommender in authentic settings, we propose a recommendation-level guided denoiser (RGD) reconstruction loss.

Definition 35 (Recommendation-level Guided Loss)**.** *Let i be the attacked item with (x, x^*)-pair of clean and perturbed images, let $\widetilde{x}$ the image denoised by AiD, then the RGD loss is defined as follows:*

$$\mathcal{L}_{RGD} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \left(\hat{s}_{ui}(\varphi) - \hat{s}_{ui}(\widetilde{\varphi}, u, i) \right)^2 \tag{6.11}$$

At this point, we can define the final AiD loss function.

Definition 36 (AiD Loss)**.** *Let $\mathcal{L}_{HGD}$ the high-level guided loss function and $\mathcal{L}_{RGD}$ the recommendation-level guided loss, we defined the AiD loss function as follows:*

$$\mathcal{L}_{AiD} = \mathcal{L}_{HGD} + \eta \mathcal{L}_{RGD} \tag{6.12}$$

where η is a coefficient to control the impact of $\mathcal{L}_{RGD}$.

Note that AiD is an unsupervised model, since the ground truth labels of product images are not needed in its training process.

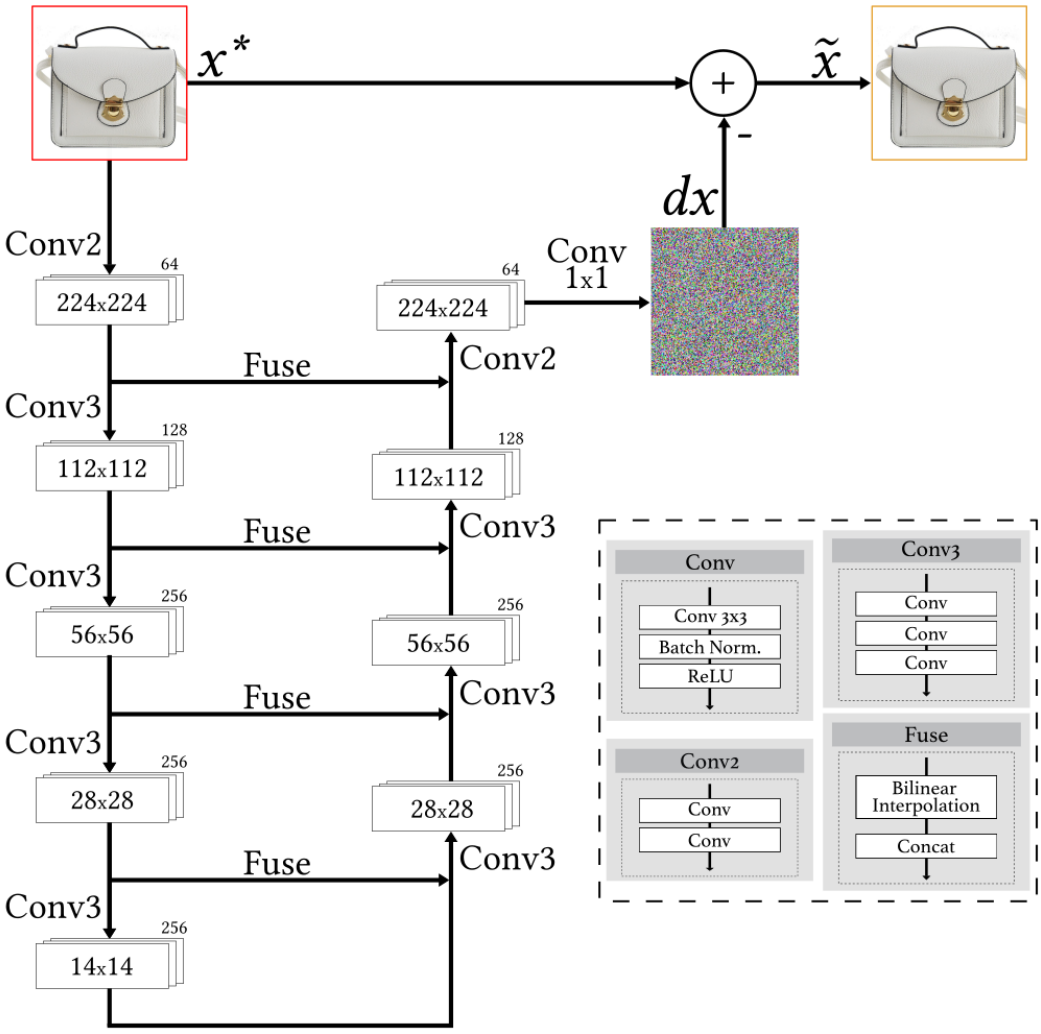


Fig. 6.2 The detail of AiD Architecture.
The **Conv** operational block is reported in the right part of the figure.

6.3.3 Training Procedure

After having introduced the denoiser architecture and its loss function, the AiD optimization problem is defined as follows:

$$\arg\min_{\Omega} \mathcal{L}_{AiD} = \arg\min_{\Omega} \left(\mathcal{L}_{HGD} + \eta \mathcal{L}_{RGD} \right) \tag{6.13}$$

A general overview of the training algorithm and back-propagation of the errors is shown in Figure 6.3. The pseudocode of the algorithm used to train AiD is presented in Algorithm 3, where μ is the learning rate used to train the denoiser, and $\mathcal{D}^*$ is

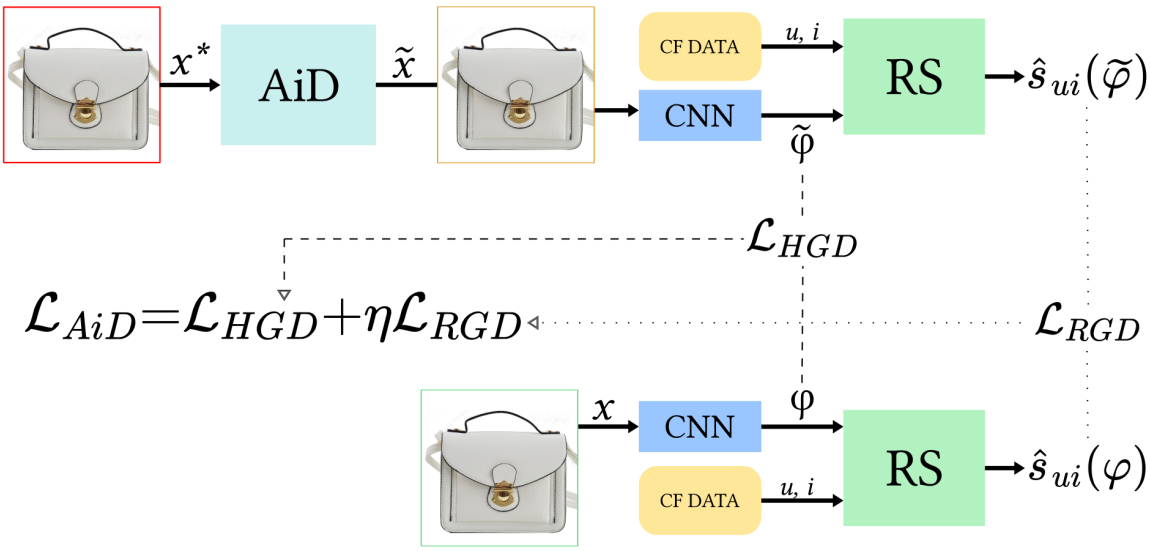


Fig. 6.3 Graphical Overview of AiD Training Algorithm.

the dataset containing the all adversarial images used to train, validate, and test the denoiser.

6.4 Experiments

Here, we present experimental settings and the discussion of the results.

6.4.1 Settings

In this section, we first introduce the real-world datasets and the procedure to create the adversarial images used to train and evaluate AiD. Then, we present the visual recommenders and the evaluation metrics. Finally, we report experimental details.

Datasets

Recommendation Datasets. We test our defensive method on the following datasets commonly utilized to evaluate VRSs.

- `Amazon Boys & Girls` [114, 159]) is an Amazon.com fashion dataset containing implicit feedback towards clothing articles. as suggested by He and McAuley [113, 114], we filter with the 5-core technique by removing the users, as well as, the items with less than five feedbacks.

Algorithm 3 Training of AiD

Input: CF data $\mathbf{S}$, Dataset of Adv. Images $\mathcal{D}_T^*$, $\mathcal{D}_V^*$.
Initial Parameters: Θ and Φ (fixed), Ω (trainable)
Output: Ω for AiD (trainable)
for epoch $= 1, ..., N_{ep}$ **do**
    $ValidLoss \leftarrow -\infty$, $\Omega_{BEST} \leftarrow \Omega$
    for $x^* \in \mathcal{D}_T^*$ **do**
        // Compute AiD Loss
        $x \leftarrow x^*$ corresponding clean image
        $\widetilde{x} \leftarrow d(x^*)$
        $\widetilde{\varphi}, \varphi \leftarrow f(\widetilde{x}), f(x)$
        $\mathcal{L}_{AiD} \leftarrow ||\varphi - \widetilde{\varphi}|| + \eta \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \left(\hat{s}_{ui}(\varphi) - \hat{s}_{ui}(\widetilde{\varphi}, u, i) \right)^2$
        // Compute Ω Gradients and Perform SGD-updates
        $g_\Omega \leftarrow {}^{\partial \mathcal{L}_{AiD}(\Omega)}/_{\partial \Omega}$
        $\Omega \leftarrow \Omega + \mu g_\Omega$
    // Compute Validation Loss on $\mathcal{D}_V^*$
    $EpValidLoss \leftarrow 0$
    for $x^* \in \mathcal{D}_T^*$ **do**
        $EpValidLoss \leftarrow EpValidLoss + \mathcal{L}_{AiD}(x^*)$
    $EpValidLoss \leftarrow {}^{EpValidLoss}/_{|\mathcal{D}_V^*|}$
    if $\mathcal{L}_{AiD}(\mathcal{D}_V^*) \leq$ ValidLoss **then**
        ValidLoss $\leftarrow EpValidLoss$
        $\Omega_{BEST} \leftarrow \Omega$
$\Omega \leftarrow \Omega_{BEST}$

- `Amazon Men` [159, 113, 114] is another popular dataset containing men's clothing from the Amazon.com category "Clothing, Shoes and Jewelry". As in [113, 114], we use the 5-core filtering.

- `Pinterest` [97, 116] collects images and interaction data from the homonym social platform. After having downloaded the item images still available on the platform, we apply 5-core on users.

For each image related to an item in the dataset, we have extracted high-level visual features with a pre-trained ResNet50 [112]. We split the dataset into training, validation, and test sets by adopting the *leave-one-out protocol* using the temporal split for `Amazon Boys & Girls` and `Amazon Men`, while random split from `Pinterest`. Table 6.1 shows the statistics of our preprocessed datasets.

Table 6.1 Statistics of the three datasets used to test AiD.

Dataset	$\mathcal{U}$	$\mathcal{I}$	S
`Amazon Boys&Girls`	1425	4507	9213
`Amazon Men`	16278	31750	113106
`Pinterest`	30375	19976	395418

Table 6.2 Adversarial images generated by different adversarial attack methods (T: iteration, ϵ: perturbation budget).

Data	**Attack**	T	ϵ
Train $\mathcal{D}_T^*$	`WB-SIGN`	1, 4, 8	rnd([1,16])
	`WB-INSA`	1, 4, 8	rnd([1,16])
Valid $\mathcal{D}_V^*$	`WB-SIGN`	1, 2, 4	rnd([1,16])
	`WB-INSA`	1, 2, 4	rnd([1,16])
Test $\mathcal{D}_\tau^*$	`BB-TAaMR`	1, 4, 8	4, 8, 16
	`WB-SIGN`	1, 4, 8	4, 8, 16
	`WB-INSA`	1, 4, 8	4, 8, 16

rnd($\cdot$) uniform sample of one integert.

Adversarial Image Datasets For each dataset, we prepared the set of adversarial images ($\mathcal{D}$) running several combinations of adversarial methods on 200 randomly extracted items (80:10:10 are the percentage of target items put into the train, validation, and test set). Inspired by the procedure used by Liao et al. [149], Table 6.2 reports all the combinations of adversarial attacks used to build the training ($\mathcal{D}_T^*$), validation ($\mathcal{D}_V^*$), and test ($\mathcal{D}_\tau^*$) sets. Note that, to prepare the test set (whose attack effects on recommendations are discussed in Section 6.4.2), we use also `BB-TAaMR` with the following most popular "target" categories (p): "Running Shoe" for the `Amazon` datasets, and "website, website, internet site, site" for `Pinterest`.

Recommenders

We test two standard visual-based recommenders:

- **VBPR** (Visual Bayesian Personalized Ranking from Implicit Feedback) [114] a MF-based model integrating the product visual features in the predicted preference score function as follows:

$$\hat{s}_{ui} = p_u^T q_i + \theta_u^T (\mathbf{E}\varphi_i) + \beta_{ui} \tag{6.14}$$

where p_u and $q_i \in \mathbb{Q}^{|\mathcal{I}| \times h}$, are the user and item embedding vectors extracted from the low-rank matrices $\mathbb{P}$ and $\mathbb{Q}$ with the number of factors h set to be $<< |\mathcal{U}|, |\mathcal{I}|$, $\theta_u \in \Theta^{|\mathcal{U}| \times v}$ is a visual-oriented latent vector of user u, $\mathbf{E}$ is a matrix to project φ_i into the h-space, and β_{ui} stands for the sum of the user, item, and global visual biases.

- **AMR** (Adversarial Multimedia Recommendation) [209] is an extension of VBPR that integrates the adversarial training procedure proposed by He et al. [115] presented in Section 6.2.3. Apart from the different training procedures, the score prediction function is the same as VBPR (see Equation (6.14)).

Evaluation

To analyze whether the denoiser is adequate, we evaluate the variations in the predicted preference scores and the top-K recommendation lists. Additionally, since the AiD-defended VRS must be valid without adversaries, we measure overall recommendation metrics.

Attack Evaluation To evaluate the adversarial attacks according to the capacity of increasing the predicted preference score, we firstly define the average prediction shift.

Definition 37 (Prediction Shift (PS))**.** *The Prediction Shift (*PS*) measures the mean variation of the preference scores across all the attacked items as follows:*

$$\text{PS} = \frac{1}{|\mathcal{D}_\tau^*|} \sum_{j \in \mathcal{D}_\tau^*} \Big(\hat{s}_{uj}(x^*) - \hat{s}_{uj}(x) \Big) \tag{6.15}$$

where $\hat{s}_{uj}(x)$ is the score predicted on the authentic image associated with the item j against which the adversary has performed an attack — whose altered predicted score is $\hat{s}_{uj}(x^*)$.

To evaluate the attack effects of recommendation ranking, we start by defining the Attack Hit Ratio (aHR@K) as in [67].

Definition 38 (Attack Hit Ratio (aHR@K))**.** *Let* $\text{attack}_{\text{hit}}\text{@K}(j, u)$ *be a hit function that is 1 when the target item is in the top-K list of the user u, 0 otherwise, then* aHR@K *is defined as:*

$$\text{aHR@K} = \frac{1}{|\mathcal{D}_\tau^*|} \sum_{j \in \mathcal{D}_\tau^*} \frac{1}{|\mathcal{U}|} \sum_{u \in |\mathcal{U}|} \text{attack}_{\text{hit}}\text{@K}(j, u) \tag{6.16}$$

where $j \in \mathcal{D}_{\tau}^{*}$ *indicates that* $\text{attack}_{\text{hit}}$@K *is measured on a target item whose image has been adversarially perturbed.*

Then, we introduce a novel measure, named Ranking Robustness.

Definition 39 (Ranking Robustness at K (RR@K))**.** *Let* aHR@K_{bef} *and* aHR@K_{aft} *be the attack hit ratios measured before and after the attack, respectively, and let*

$$\Delta\text{aHR@K} = \frac{\text{aHR@K}_{aft} - \text{aHR@K}_{bef}}{\text{aHR@K}_{bef}} \tag{6.17}$$

be the difference ratio, then RR@K *is defined as follows:*

$$\text{RR@K} = \left| \frac{\Delta\text{aHR@K}^{w}}{\Delta\text{aHR@K}^{wo}} \right| \tag{6.18}$$

where $\Delta\text{aHR@K}^{w}$ *and* $\Delta\text{aHR@K}^{wo}$ *are measured when the VRS is protected with and without AiD.*

RR@K $\simeq 0$ means that AiD has reached optimal performance, RR@K $\simeq 1$ is the scenario where AiD does not impact the attacks' efficacy, and RR@K $>> 1$ is the awful situation where the AiD could have considerably impacted the presence of target items in the top-K lists.

Recommendation Evaluation We study accuracy and beyond-accuracy metrics on top-K recommendation performance. For accuracy, we measure the recall (Rec@K), that accounts for a fraction of test items that are correctly suggested in the top-K lists, and the normalized discounted cumulative gain (nDCG@K), that analyzes the ranking position of correctly recommended items by assigning higher relevance scores with hits in top positions. To evaluate the novelty and diversity, we measure the fraction of items covered in the catalog ($iCov@k$) and the expected free discovery (EFD@K). In particular, EFD@K estimates the capacity to recommend diverse items [213]. Finally, for what regards the study on the effects of popularity bias, we adopt the metrics used by [2]: the average recommendation popularity (ARP) that calculates the popularity of the recommended items in each list, the average percentage of long-tail items (ACLT), and the average coverage of long-tail items (APLT), that estimates the exposure of long-tail items in the entire recommendations. As suggested by Abdollahpouri et al. [2], we use the 80:20-split in which the short-head are the top 20% of items by popularity, and the long-tail ones are the last 80%. As in Definition 39, we use the apices w and wo to indicate the metric values measured with and without a VRS protected by AiD.

Implementation Details.

In the first stage, we train each visual recommender with model parameters initialized by a Gaussian distribution with a mean of 0 and standard deviation of 0.01 as in [61]. We search the best-performing recommender with respect to Rec@50 by exploring their learning rate in $\{0.0001, 0.001, 0.01\}$ and the regularization coefficients in $\{0.00001, 0.001\}$, and fixing the number of training epochs to 100, the batch size to 256, and the number of latent factor (h) to 128. The adversarial epochs used for training AMR are 50 (performed after the initial 50 epochs with standard VBPR training) with $\lambda = 1$ and $\epsilon = 1$ (see Definition 33 for further details). AiD is trained for 100 epochs. We set $\eta = 0$ for the first 50 epochs to allow the denoiser to focus on the high-level guided reconstruction. Then, we train the denoiser for 50 epochs, fixing $\eta = 1$, for learning how to preserve the recommendation-level quality. We set the batch size to 16, and, following the suggestion by Liao et al. [149], we train the denoiser with the Adam optimizer with $\mu = 0.001$. We perform our experiments using the ELLIOT reproducibility framework [18] by releasing our configuration files.

6.4.2 Results and Discussion

In this section, we perform, analyze and discuss the experimental results with the aim to answer the following research questions:

RQ1 Can Adversarial Image Denoiser (AiD) reduce the effectiveness of adversarial attacks? How is it performing with respect to the state-of-the-art adversarially trained model (i.e., AMR) and, if possible, improving its robustness?

RQ2 Is the defensive strategy robust when adversaries increase the perturbation budget (ϵ) and the number of steps (T)?

RQ3 How much AiD is able to reduce the impact of adversarial attacks on top-K recommendation lists?

RQ4 How much AiD application is affecting the overall accuracy and beyond-accuracy performance?

Analysis of PS **(RQ1)**

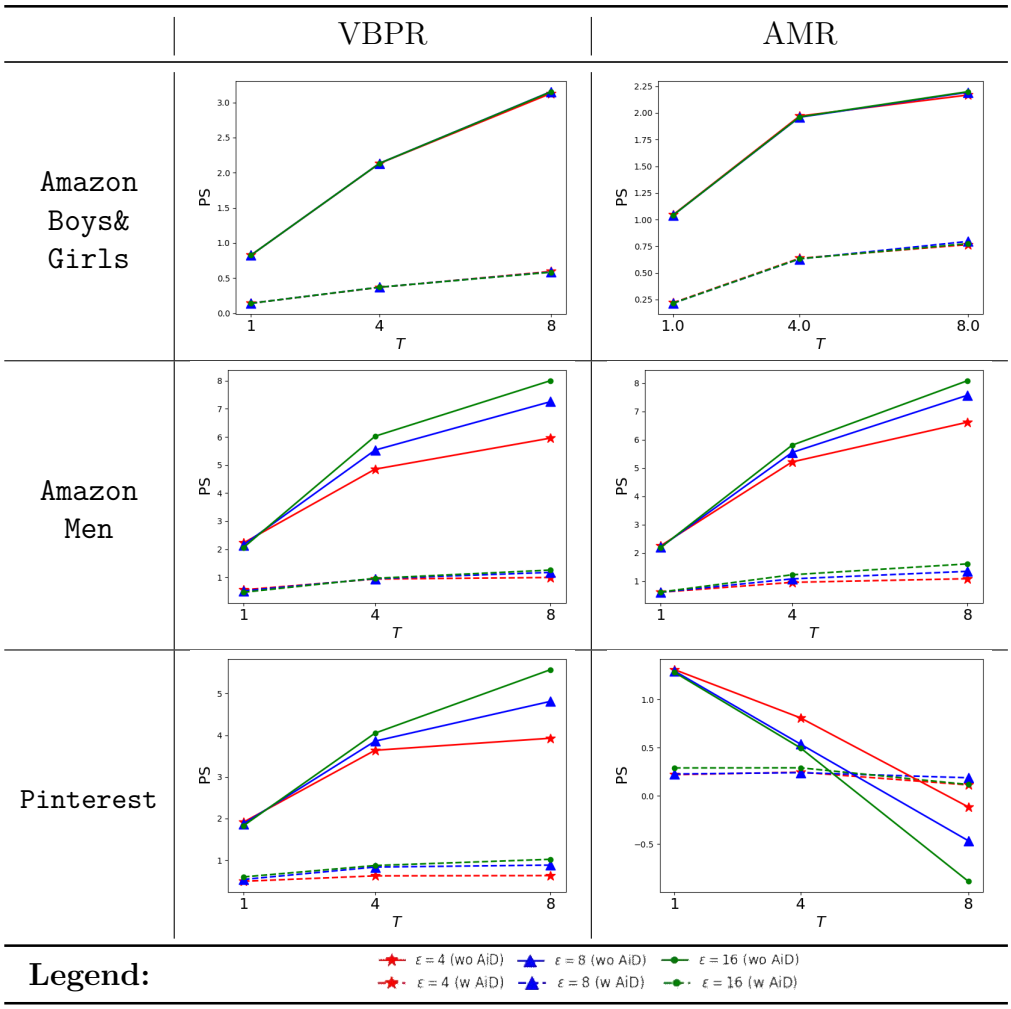
Here, we compare prediction shifts with the most human-imperceptible budget ($\epsilon = 4$) and a single iteration ($T = 1$). The results are listed in Table 6.3. We start by verifying that black-box attacks are less effective than white-box ones since, as expected, `BB`

Table 6.3 Prediction Shift measured on ($\epsilon = 4, T = 1$)-attacks without and with the use of AiD. We bold values when AiD is effective.

Dataset	**Model**	**Attack**	PS^{wo}	PS^{w}
Amazon Boys& Girls	VBPR	BB-TAaMR	-0.1437	**0.0507**
		WB-INSA	0.8250	**0.1410**
		WB-SIGN	1.8466	**1.2668**
	AMR	BB-TAaMR	0.4643	0.6648
		WB-INSA	1.0432	**0.2193**
		WB-SIGN	1.3349	**1.1183**
Amazon Men	VBPR	BB-TAaMR	-0.1072	0.1105
		WB-INSA	2.2217	**0.5560**
		WB-SIGN	2.2413	**1.0005**
	AMR	BB-TAaMR	-0.0803	**-0.0423**
		WB-INSA	2.2418	**0.6057**
		WB-SIGN	2.5066	**1.0969**
Pinterest	VBPR	BB-TAaMR	0.4784	**0.1729**
		WB-INSA	1.9113	**0.4931**
		WB-SIGN	1.8929	**0.6434**
	AMR	BB-TAaMR	0.7163	**0.1470**
		WB-INSA	1.3108	**0.2205**
		WB-SIGN	1.2817	**0.3345**

attacks have a complete absence of knowledge on the model and dataset. Indeed, contrary to BB, WB attacks cause a vast increase in predicted preference scores in every tested scenario (e.g., $PS^{wo} > +1$). At this point, we analyze the capacity of AiD in protecting VBPR. It can be seen that the use of AiD has been effective in reducing the average prediction shifts for nearly all combinations of black-box and white-box attacks and VBPR. For instance, PS^{w} is always reduced by more than three times for each WB-INSA attack independently of the datasets (e.g., $0.1410 < 0.8250$; $0.5560 < 2.2217$; and $0.4931 < 1.9113$ from the top of the table to the bottom). Extending the analysis to the integration of AiD with the adversarially trained visual recommender (i.e., AMR), the results in Table 6.3 widely validate the AiD's quality in reducing the adversaries' goal of altering the predicted preference scores. As an example, PS moves from 0.7163 to 0.1470, and from 1.2817 to 0.3345 on Pinterest subjected to BB-TAaMR and WB-SIGN. These results endorse that AiD is an effective defensive solution, even when used together with adversarially trained recommenders.

Table 6.4 Prediction Shift (PS) of the `WB-INSA` attack by varying the budget ($\epsilon \in \{4, 8, 16\}$) and the number of iterations ($T \in \{1, 4, 8\}$).



Analysis of PS when varying ϵ and T (RQ2)

Table 6.4 presents six plots that show PS^{wo} and PS^{w} when varying the number of steps $T \in \{1, 4, 8\}$ and the perturbations budget $\epsilon \in \{4, 8, 16\}$ only for the `WB-INSA` attack performed against both recommenders being the `WW`-attack with the lowest PS^{w} values. First, analyzing the continuous lines, we get evidence that PS^{wo} gets a considerable boost in the absence of the denoiser with empowered adversaries (bigger T or ϵ). The only exception is for the attacks against AMR trained on the `Pinterest` dataset, where the decrease of the attack effects can be explained by the fact that the adversarial training might have influenced the efficacy of more potent attacks. However, the application of the denoiser (dotted lines) has effectively intercepted the attempts of stronger adversaries by always showing very low prediction shifts. For instance, it can

Table 6.5 Analysis of top-50 ranking-aware performance of for the VRSs without and with the use of AiD.

Dataset	**Model**	**Attack**	ΔaHR^{wo}	ΔaHR^{w}	RR
Amazon Boys& Girls	VBPR	BB-TAaMR	-0.3011	-0.0899	**0.2986**
		WB-INSA	0.9677	0.1461	**0.1509**
		WB-SIGN	2.7419	1.8792	**0.6854**
	AMR	BB-TAaMR	-0.3204	-0.2179	**0.6799**
		WB-INSA	-0.4972	-0.2542	**0.5112**
		WB-SIGN	-0.8149	-0.5726	**0.7027**
Amazon Men	VBPR	BB-TAaMR	-0.1552	-0.3777	2.4335
		WB-INSA	1.8702	-0.1739	**0.0930**
		WB-SIGN	1.6921	0.1304	**0.0771**
	AMR	BB-TAaMR	55.6936	-0.2752	**0.0049**
		WB-INSA	79.8543	0.2141	**0.0027**
		WB-SIGN	81.6890	0.4327	**0.0053**
Pinterest	VBPR	BB-TAaMR	0.8272	-0.1413	**0.1709**
		WB-INSA	0.2057	-0.2310	1.1232
		WB-SIGN	0.0543	-0.1809	3.3317
	AMR	BB-TAaMR	0.1732	-0.0218	**0.1260**
		WB-INSA	0.6665	0.0087	**0.0131**
		WB-SIGN	0.6464	0.1945	**0.3009**

be noted that while PS^{wo} increase from values close to 1 to higher than 3 for VBPR trained on `Amazon Boys` & `Girls`, PS^{w} always remains less than 1. The same efficient behavior can also be noted on the other plots, where AiD guarantees consistently low variations of the predicted scores under stronger and stronger attacks.

Analysis of Ranking-based Measures (RQ3)

Rising the average position of target items into a high recommendation position is another adversary's goal strictly pursued with the increase of predicted scores. Table 6.5 reports the results of the ranking-aware metrics presented in Section 6.4.1 measured on top-50 recommendation lists. We can see that applying the proposed denoising approach has been adequate in most of the tested scenarios. Indeed, the fact that the RR values are mostly smaller than 1 in any attack scenario demonstrates that the presence of the AiD has reduced the adversaries' capability in pushing the target items in higher recommendation positions. Additionally, it is interesting to observe that the only three scenarios in which the RR is higher than 1 are related to cases where the adversarial attacks were not very powerful also in the not-defended setting. In these

Table 6.6 Overall recommendation performance measured in no adversarial settings without and with the use of AiD. R.V.measures the percentage of variation between the metric values measured on not-defended and defended recommender. We put in bold the positive **R.V.** to represent an improvement of the metric value.

Dataset	Model	Def.	Accuracy				Beyond-accuracy									
			Rec		nDCG		iCov		EFD		ACLT		APLT		ARP	
			@20	@50	@20	@50	@20	@50	@20	@50	@20	@50	@20	@50	@20	@50
Amazon Boys& Girls	VBPR	No	0.0337	0.0653	0.0121	0.0182	0.8005	1.0067	0.0197	0.0162	8.0821	20.8618	0.4041	0.4172	2.4575	2.1493
		AiD	0.0295	0.0667	0.0118	0.0190	0.8030	1.0104	0.0195	0.0170	8.1516	21.0793	0.4076	0.4216	2.1840	1.9734
		R.V.	-12.46	**2.14**	-2.48	**4.40**	**0.30**	**0.37**	-1.02	**4.94**	**0.86**	**1.04**	**0.87**	**1.05**	-11.13	-8.18
	AMR	No	0.0316	0.0582	0.0113	0.0165	0.7630	0.9891	0.0191	0.0152	8.1277	21.0281	0.4064	0.4206	1.7296	1.6292
		AiD	0.0316	0.0611	0.0111	0.0169	0.7564	0.9896	0.0186	0.0154	8.1144	21.0007	0.4057	0.4200	1.7359	1.6316
		R.V.	**0.00**	**4.98**	-1.77	**2.42**	-0.87	**0.04**	-2.62	**1.32**	-0.16	-0.13	-0.17	-0.14	**0.36**	**0.15**
Amazon Men	VBPR	No	0.0144	0.0283	0.0056	0.0083	0.5941	0.7975	0.0100	0.0083	7.1428	19.1679	0.3571	0.3834	10.9857	9.0988
		AiD	0.0139	0.0270	0.0054	0.0080	0.5857	0.7845	0.0099	0.0080	7.5198	19.9042	0.3760	0.3981	10.5176	8.7409
		R.V.	-3.47	-4.59	-3.57	-3.61	-1.41	-1.64	-1.00	-3.61	**5.28**	**3.84**	**5.29**	**3.83**	-4.26	-3.93
	AMR	No	0.0081	0.0187	0.0032	0.0053	0.4035	0.6066	0.0057	0.0053	8.6131	22.5334	0.4307	0.4507	8.7773	7.2129
		AiD	0.0079	0.0168	0.0031	0.0049	0.5630	0.7654	0.0060	0.0052	10.0850	25.7561	0.5042	0.5151	5.0164	4.4566
		R.V.	-2.47	-10.16	-3.13	-7.55	**39.52**	**26.16**	**5.26**	-1.89	**17.09**	**14.30**	**17.07**	**14.29**	-42.85	-38.21
Pinterest	VBPR	No	0.0597	0.1180	0.0236	0.0351	0.7664	0.9166	0.0467	0.0381	4.1532	11.2491	0.2077	0.2250	21.3674	20.4271
		AiD	0.0479	0.1000	0.0180	0.0282	0.7173	0.8805	0.0358	0.0307	4.5285	11.9408	0.2264	0.2388	20.2717	19.6827
		R.V.	-19.77	-15.25	-23.73	-19.66	-6.41	-3.94	-23.34	-19.42	**9.04**	**6.15**	**9.00**	**6.13**	-5.13	-3.64
	AMR	No	0.0301	0.0676	0.0111	0.0184	0.6455	0.8300	0.0224	0.0203	5.6196	14.4269	0.2810	0.2885	17.9860	17.8545
		AiD	0.0289	0.0627	0.0106	0.0172	0.6454	0.8230	0.0215	0.0190	5.8936	14.8948	0.2947	0.2979	17.6031	17.5617
		R.V.	-3.99	-7.25	-4.50	-6.52	-0.02	-0.84	-4.02	-6.40	**4.88**	**3.24**	**4.88**	**3.26**	-2.13	-1.64

contexts, we note that the ΔaHR^{wo} and ΔaHR^{w} metric values are very close to 0 (i.e., -0.1552 and -0.3777 for <`Amazon Men`, VBPR, `BB-TAaMR` >, 0.2057 and -0.2310 for <`Pinterest`, VBPR, `WB-INSA` >, 0.0543 and -0.1809 for <`Pinterest`, VBPR, `WB-SIGN` >). We can summarize that AiD effectively reduces the adversaries' impact in varying the predicted preference scores and, as shown in this paragraph, preserving the target items' position in the not-attacked recommendation lists.

Recommendation Performance (RQ4)

Here, we study the accuracy and beyond-accuracy recommendation performance measured on each visual recommender when protected (or not) by the proposed adversarial denoiser. Table 6.6 shows the top-K recommendation performance on the three datasets, where $K \in \{20, 50\}$. Globally, the results show that the integrating of AiD allows preserving a consistent part of the accuracy and beyond-accuracy of the correspondent not-defended recommender (e.g., VBPR with AiD vs. VBPR without AiD). Indeed, analyzing the accuracy measures, we see that Rec and nDCG can, on average, suffer from small reductions (i.e., $\simeq 2-3\%$). For instance, the **R.V.** values on VBPR trained on `Pinterest` are the worst ones (e.g., **R.V.** = -19.77% on Rec@20), which might be explained by the fact the images of `Pinterest` are more visually complicated to be denoised if compared with the clothing items of the Amazon datasets that are placed on the white background. However, results also present a best-case scenario where the accuracy can even be improved (e.g., **R.V.** of +4.98% recorded on

Rec@50 for AMR trained on `Amazon Men`). Similarly, Table 6.6 shows that the impact of AiD on coverage and diversity measures is quite limited. For instance, iCov@20 receives a positive relative variation of 3.3583% when averaged across all the datasets and models.

Finally, we explore the effects of AiD on the metrics used to study the effects of popularity bias on the produced recommendation lists (ARP, ACLT, and APLT). The exposure metrics (i.e., ACLT and ACLT) reveals that the AiD defensive algorithm is doing a much better job of exposing items across the long-tail part of the catalog. Indeed, both metrics get an average **R.V.** higher than 4% in any setting (on both top-20 and 50 recommendation lists). This exposure benefit is also confirmed by a steady average **R.V.**of ARP of -9%. We can conclude that AiD effectively robustifies the visual recommenders against adversarial attacks by preserving the overall recommendation performance measured in not-attacked settings.

6.5 Summary

This chapter has investigated the vulnerability of visual-based recommender systems (VRS) to adversarial attacks— human indistinguishable perturbed product images uploaded on a recommender to maliciously change the rank of target products— by proposing a novel defensive solution. We have proposed a denoiser network, named Adversarial Image Denoiser (AiD), to be placed before the convolutional neural network used to extrapolate visual image features trained to learn how to remove the adversarial noise on input images guided both by a feature- and recommendation-aware reconstruction loss. We have investigated the defense performance on three real-world datasets and two popular visual recommender models, one of which implements the state-of-the-art defensive solution (i.e., adversarial training) under three attack strategies (i.e., one black-box and two white-box). The experiments confirm that AiD is an effective solution for protecting visual recommender models against the set of tested attacks, reducing their effectiveness in varying the predicted preference scores and the target items' positions in the recommendation lists. Additionally, we have verified that the integration of this defense does not worsen the overall accuracy and beyond-accuracy recommendation performance, with effects that have been advantageous in some cases. We plan to extend this defense strategy to identifying possible stronger adversarial attacks that might break the AiD defensive power in order to design increasingly resistant defense for the sake of users, sellers, and platforms.

Chapter 7

Iterative Adversarial Perturbations on Model Parameters

Considering the parameters' instability to adversarial perturbation on model-based RSs, how vulnerable are the parameters to iterative gradient-based adversarial methods?

Is Adversarial Personalized Ranking effective in robustifying the model against iterative methods?

RSs have attained exceptional performance in learning users' preferences and finding the most suitable products. Recent advances in adversarial machine learning (AML) in computer vision have raised recommenders' security interests. It has been demonstrated that widely adopted model-based recommenders, e.g., BPR-MF, are not robust to adversarial perturbations added on the learned parameters, e.g., users' embeddings, which can cause a drastic reduction of recommendation accuracy (see Section 2.3.2). However, the state-of-the-art adversarial method, named the fast gradient sign method (FGSM), builds the perturbation with a single-step procedure. This chapter extends the FGSM method, proposing multi-step adversarial perturbation (`MSAP`) procedures to study the recommenders' robustness under powerful methods. Letting fixed the perturbation magnitude, we illustrate that `MSAP` is much more harmful than FGSM in corrupting the recommendation performance of BPR-MF. Then, we assess the `MSAP` efficacy on a robustified version of BPR-MF, i.e., AMF. Finally, we analyze the variations of fairness measurements on each perturbed recommender. Code and data are available [1].

[1] https://github.com/sisinflab/MSAP

7.1 Introduction

[115] have proposed the pioneering work of AML for RSs in the AML application of *Adversarial Perturbations of Model Parameters* as shown in Section 2.3.2. Starting from the authors' clarification that attacks and defenses should be treated differently in the CV and RS tasks since image data are continuous-valued matrices, while recommender data are discrete interactions (0/1 feedback); they have investigated adversarial methods to perturb the model parameters, e.g., the embedding matrices of matrix-factorization (MF) models. They discovered that the fast gradient sign method (FGSM) [101], a *single-step adversarial perturbation* procedure, leads to five times larger deterioration of recommendation accuracy than the one caused by random variation. This finding shows the weaknesses of model-based recommenders in learning embeddings that will cause drastic performance degradation when subjected to small changes. For instance, this change can be caused when new users, or items, are added to the system. Furthermore, they successfully applied an *adversarial training* procedure [101] on BPR-MF, named AMF, demonstrating more robust RS performance under FGSM perturbations. These techniques have been tested on multimedia recommendation systems [209], deep RSs [234, 235], and tensor factorization approaches [58].

In this chapter, inspired by the evidence in the CV domain that iterative attacks are more effective than single-step ones [142], we present two *multi-step adversarial perturbation* (`MSAP`) techniques, namely primary iterative method (BIM) and projected gradient descent (PGD), applied on the embeddings of two state-of-the-art MF models [115, 188]. Our idea is to investigate whether the attack empowerment obtained in CV settings are valid in RS tasks to confirm the presence of minimal perturbations that might cause an enormous worsening of the model stability/robustness. Particularly, we make the following contributions:

- proposes a novel attack method, named Multi-Step Adversarial Perturbation (`MSAP`), to study whether its impact in degrading the quality of the system with respect to accuracy and beyond-accuracy evaluation measures when compared to single-step ones (i.e., the attack by He et al. [115]);
- test the state-of-the-art robustification procedure of model-based recommenders (APR presented in Section 2.3.2) against the presented multi-step noise;
- study whether adversarial perturbations and in particular `MSAP` can significantly impact the observed fairness of recommender models.

To this end, we evaluated the impact of the proposed adversarial iterative strategies for item recommendation task against two standard model-based collaborative recommenders, i.e., BPR-MF [188] and its adversarial defended version AMF [115], on two well-recognized recommender datasets, i.e., `ML-1M` and `LastFM`. Overall, the considered attacks highlight the necessity to investigate new defensive measures to limit their effectiveness in reducing recommendation performance (accuracy, beyond-accuracy, and fairness).

The rest of the chapter is organized as follows. In Section 7.2, we formalize the problem of iterative adversarial perturbations, and in Section 7.3 we present the setting and the results of our empirical evaluation of the proposed method. Then, in Section 7.4 reviews the related work before presenting the contribution summary in Section 7.5.

7.2 Method

In this section, we describe the foundations of a personalized matrix factorization (MF) recommender model. Then, we recapitulate the baseline single-step adversarial perturbation before defining the multi-step strategies.

7.2.1 Personalized Recommenders via MF

The recommendation problem is the task of estimating a preference prediction function $s(u,i)$ that maximizes the utility of the user $u \in \mathcal{U}$ in getting the item $i \in \mathcal{I}$ recommended by the RS, where Before we dive into the description of the MF model, we recap the notations:

- $\mathcal{U}$ and $\mathcal{I}$ are the set of users and items, respectively;
- $\mathbf{P}$: the matrix of *user* embeddings, where $\mathbf{p}_u$ is the embedding vector associated to the user u;
- $\mathbf{Q}$: the matrix of *item* embeddings, where $\mathbf{q}_i$ is the embedding vector associated to the item i;
- $\mathbf{\Theta}$: the set of model parameters ($\mathbf{\Theta} = \{\mathbf{P}, \mathbf{Q}\}$);
- $\mathbf{\Delta}$: the set of adversarial perturbation on model parameters ($\mathbf{\Delta} = \{\delta_{\mathbf{P}}, \delta_{\mathbf{Q}}\}$);
- $\mathcal{L}$: the loss function of the recommender model.

The main intuition behind the MF model is to compute the preference score $\hat{s}(i|u)$ as the dot product between the user's embedding ($\mathbf{p}_u$) and the item's embedding ($\mathbf{q}_i$). The model parameters are learned by solving the optimization problem in the following general form:

$$\underset{\boldsymbol{\Theta}}{\operatorname{argmin}}\, \mathcal{L}(\boldsymbol{\Theta}) \tag{7.1}$$

The state-of-the-art approach to produce personalized rankings is Bayesian personalized ranking (BPR) [188]. The idea is to reduce the ranking problem to a pairwise learning one where, for each user, the score of interacted items has to be higher than non-interacted ones.

7.2.2 Adversarial Perturbation of Model Parameters

The main intuition behind an adversarial perturbation method is to generate minimum perturbations ($\boldsymbol{\Delta}^{adv}$) capable of undermining the learning objective of the learning model. The adversary's goal is to maximize Equation (7.1), under a minimal-norm constraint

$$\Delta^{adv} \leftarrow \underset{\boldsymbol{\Delta}_0, ||\boldsymbol{\Delta}_0|| \leq \epsilon}{\operatorname{argmax}} \mathcal{L}(\boldsymbol{\Theta} + \boldsymbol{\Delta}_0) \tag{7.2}$$

where $\boldsymbol{\Delta}_0$ is the initial adversarial perturbation added to the model parameters $\boldsymbol{\Theta}$ and ϵ is the *perturbation budget* (the limit of the perturbation size).

Equations (7.1) and (7.2) can be unified in the following *minimax* problem:

$$\arg \min_{\boldsymbol{\Theta}} \max_{\boldsymbol{\Delta}_0, ||\boldsymbol{\Delta}_0|| \leq \epsilon} \mathcal{L}(\boldsymbol{\Theta} + \boldsymbol{\Delta}_0) \tag{7.3}$$

in which two opposite players play an **adversarial minimax** game, where the adversary tries to maximize the likelihood of its success while the ML model tries to minimize the risk. This minimax game is the main characteristic of tasks related to AML research [212].

Fast Gradient Sign Method (FGSM).

This perturbation strategy is the baseline *single-step adversarial perturbation* mechanism proposed by [115] to alter the recommendation task. It builds on advances made in ML research pioneered in [101] for the classification task. It approximates $\mathcal{L}$ by linearizing it around an initial zero-matrix perturbation $\boldsymbol{\Delta}_0$ and applies the max-norm constraint.

The adversarial noise $\mathbf{\Delta}^{adv}$ is

$$\mathbf{\Delta}^{adv} = \epsilon \frac{\Pi}{||\Pi||} \quad \text{where} \quad \Pi = \frac{\partial \mathcal{L}(\mathbf{\Theta} + \mathbf{\Delta}_0)}{\partial \mathbf{\Delta}_0} \tag{7.4}$$

where $||\cdot||$ is the L_2-norm. After the calculation of $\mathbf{\Delta}^{adv}$, the authors added this perturbation to the current model parameters $\mathbf{\Theta}^{adv} = \mathbf{\Theta} + \mathbf{\Delta}^{adv}$ and generated the recommendation lists with this perturbed model parameter.

Multi-Step Adversarial Perturbation (`MSAP`).

This adversarial noise generation mechanism is a straightforward extension of the single-step strategy proposed in the CV domain [142]. In particular, the authors' idea was to build an FGSM-based *multi-step* strategy and create more effective ϵ-clipped perturbations. The initial model parameters are defined as

$$\mathbf{\Theta}_0^{adv} = \mathbf{\Theta} + \mathbf{\Delta}_0 \tag{7.5}$$

Starting from this initial state of model parameters, let $Clip_{\mathbf{\Theta},\epsilon}$ be an element-wise clipping function to limit the perturbation of each original embedding value inside the $[-\epsilon, +\epsilon]$ interval, let α be the step size which is the maximum perturbation budget of each iteration, and let L be the number of iterations, the first iteration ($l = 1$) is defined by:

$$\mathbf{\Theta}_1^{adv} = Clip_{\mathbf{\Theta},\epsilon}\left\{\mathbf{\Theta}_0^{adv} + \alpha \frac{\Pi}{||\Pi||}\right\} \text{ where } \Pi = \frac{\partial \mathcal{L}(\mathbf{\Theta} + \mathbf{\Delta}_0)}{\partial \mathbf{\Delta}_0} \tag{7.6}$$

and we generalize the l-th iteration of the L-iterations multi-step adversarial perturbation as:

$$\mathbf{\Theta}_l^{adv} = Clip_{\mathbf{\Theta},\epsilon}\left\{\mathbf{\Theta}_{l-1}^{adv} + \alpha \frac{\Pi}{||\Pi||}\right\} \text{ where } \Pi = \frac{\partial \mathcal{L}(\mathbf{\Theta} + \mathbf{\Delta}_{l-1}^{adv})}{\partial \mathbf{\Delta}_{l-1}^{adv}} \tag{7.7}$$

where $l \in [1, 2, ..., L]$, $\mathbf{\Delta}_l^{adv}$ is the adversarial perturbation at the l-th iteration, and $\mathbf{\Theta}_l^{adv}$ is the sum of the original model parameters $\mathbf{\Theta}$ with the perturbation at the l-th iteration. The `MSAP` computational cost is l-times the single-step version. We considered two different `MSAP`: Basic Iterative Method (BIM) [142] and Projected Gradient Descent (PGD) [155]. The former approach initializes $\mathbf{\Delta}_0$ as matrices of zeros, with the same size as the matrix embeddings of the victim model. The latter initializes the perturbation by sampling from a uniform distribution. These different initialization make PGD more powerful than BIM in confusing CV image classifiers [33]. Since this

has not been – to the best of our knowledge – investigated in the RSs community, we chose both strategies to investigate whether such a difference between two adversarial perturbation strategies exists for the recommendation task. Note that we test our adversarial method against MF recommenders. However, it can be reproduced against any BPR optimized recommender.

7.3 Experiments

Here, we present experimental settings and the discussion of the empirical results.

7.3.1 Settings

In this section, we introduce the datasets, recommenders, evaluation measures, and reproducibility information.

Datasets

We perform `MSAP` experiments on two datasets:

- **MovieLens 1M** (`ML-1M`) [109] contains 1,000,209 ratings ($|\mathcal{F}|$) given by 6,040 users ($|\mathcal{U}|$) towards 3,706 movies ($|\mathcal{I}|$). Users' gender and movies' genres are used in the fairness evaluation.
- **LastFM-1b** (`LastFM`) [195] contains 935,875 listening events ($|\mathcal{F}|$) given by 2,847 users ($|\mathcal{U}|$) towards 33,164 authors ($|\mathcal{I}|$) stored from the online music provider `Last.fm`. Users' gender and items' artists are used for the analysis of fairness.

We employ the *leave-one-out* evaluation protocol [115], putting in the test set either the last — when that information is available (i.e., `ML-1M`)– or a random (i.e., `LastFM`) interaction, and using the rest of the recorded feedbacks to train the recommenders.

Recommender Models

We execute the experiments on two models:

- **BPR-MF** [188] is a MF recommender optimized with a pair-wise loss function (i.e., BPR). The fundamental intuition of BPR-MF is to discard not-interacted items with respect to interacted ones in order to learn a rank-based preference predictor. $\mathcal{L}_{BPR}(\mathbf{\Theta}) = \mathcal{L}(\mathbf{\Theta})$ denotes the BPR-MF loss function. Additional MF details are presented in Section 2.1.1.

- **AMF** [115] is a BPR-MF extension that includes an adversarial training procedure. The model parameters are learned with the following loss function:

$$\mathcal{L}_{AMF}(\boldsymbol{\Theta}) = \mathcal{L}_{BPR}(\boldsymbol{\Theta}) + \lambda \underbrace{\mathcal{L}_{BPR}(\boldsymbol{\Theta}^{adv})}_{\text{adversarial regularizer}} \tag{7.8}$$

  where the model parameters of the *adversarial regularizer* ($\boldsymbol{\Theta}^{adv}$) are perturbed with the single-step perturbation technique described in Equation (7.4). AMF reduces up to 88% the degrading effect of single-step perturbations on the model accuracy [115]. Additional details on the adversarial training procedure are presented in Section 2.3.

Evaluation Metrics

To verify the efficacy of `MSAP`, we evaluate the effectiveness of our methods using the following set of metrics:

- **Accuracy** The accuracy metrics used are precision (Pr@K), the fraction of suggested items relevant to the users, recall (Re@K), the average fraction of relevant recommended items, and normalized discounted cumulative gain (nDCG@K), the users' gain of a ranked list discounting the relevant predictions by their positions. Further details are presented in Section 2.1.2.

- **Beyond-Accuracy** The beyond-accuracy metrics used are: expected free discovery (EFD@K) [214], the capacity to suggest relevant long-tail products, Shannon Entropy (SE@K), the diversity of recommendations, and coverage (ICov@K), the number of recommended products.

- **Fairness** metrics are evaluated before and after `MSAP`. We explored: generalized cross-entropy (GCE) [74] that considers several possible ideal probability distributions for each user, or item, clustering. Hence, it computes the divergence of the recommendation results (by considering a specific metric, i.e., nDCG) from the ideal distributions. Consequently, GCE's value close to zero denotes the recommender's congruence with that distribution. On the other hand,MAD focuses on the absolute variation of a given metric from an ideal situation in which the recommender treats groups equally. The original formulation of MAD [246], namely MADr, considers the user and item score pairs in the recommendation results. Additionally, we considered theMAD extension proposed in [74], MADR,

in which the per-user performance values of an accuracy metric, i.e., nDCG, are considered.

Evaluation Protocol

We train the BPR-MF for 2,000 epochs. Then, we use BPR-MF's parameters at the 1,000-th epoch as the starting point to train AMF — the BPR-MF *adversarial regularized* version– as presented in [115]. We fix the perturbation budget (ϵ) to 0.5, which is the smallest perturbation experimented in [115], and set the step size α of `MSAP` to $\epsilon/4$. We employ the following parameters for both models: embedding size (h) to 64, learning rate to 0.05, λ to 1, and the batch size to 512.

7.3.2 Results and Discussion

Here, we perform experiments to answer the following research questions:

RQ1 Does MSAP outperform single-step attacks in degrading the system's quality with respect to accuracy and beyond-accuracy evaluation measures?

RQ2 Is the *adversarial regularization* of RSs still useful against the presented multi-step generated noise?

RQ3 Are adversarial perturbations, and in particular the `MSAP`, able to impact in a significant direction on the observed fairness of recommender models?

Investigating the `MSAP` Effects (RQ1-2)

To better understand the merits of the presented adversarial perturbations, we aim to answer the following questions:

- **On the perturbation side (RQ1)**: how much adversarial perturbations obtained from the single-step and the `MSAP` methods can impair the quality of the original BPR-MF model? Figures 7.1a and 7.1c compare perturbations effects on BPR-MF trained on `LastFM`.
- **On the defensive side (RQ2)**: what is the impact on the adversarial regularized version of BPR-MF, i.e., AMF? The answer can be found in Figures 7.1b and 7.1d.

Since the performance of the `MSAP` varies based on the number of iterations, firstly, we discuss and analyze the effectiveness of the presented perturbations across different

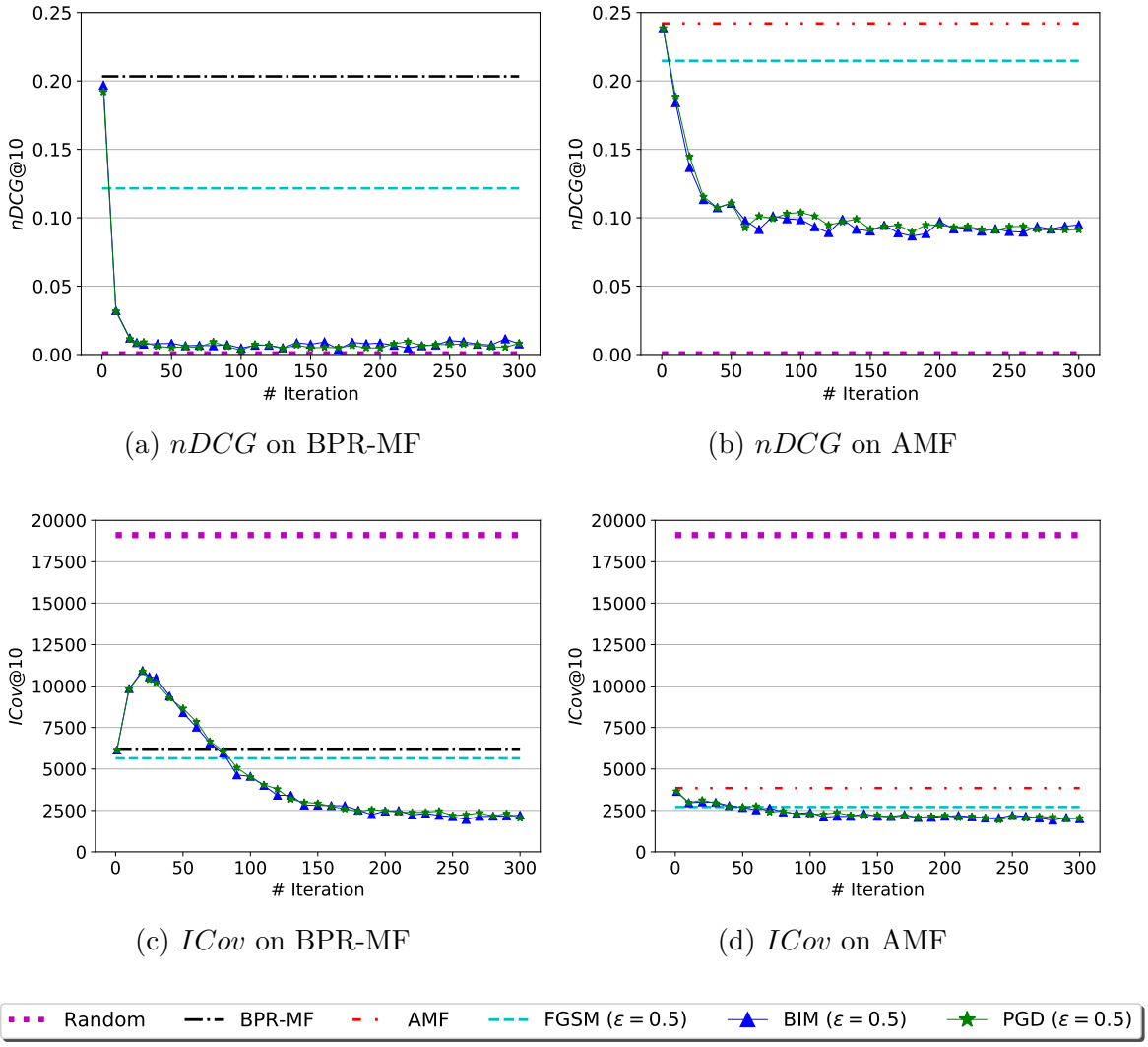
Table 7.1 Accumulated normalized values of the accuracy and beyond-accuracy metrics. We put in **bold** the lower value when the perturbation ($\epsilon = .5$) is more effective.

Model	Metric	LastFM				ML-1M			
		Initial	FGSM	BIM	PGD	Initial	FGSM	BIM	PGD
BPR-MF	*PR*	.0310	.0211	.0019	**.0018**	.0088	.0074	**.0035**	**.0035**
	RE	.3102	.2115	.0194	**.0177**	.0884	.0740	**.0353**	**.0353**
	nDCG	.2033	.1216	.0111	**.0100**	.0447	.0368	.0174	**.0172**
	EFD	.5144	.3069	.0313	**.0284**	.0977	.0791	.0355	**.0353**
	SE	11.35	11.14	**1.17**	1.21	9.63	9.16	**7.40**	7.45
	ICov	6220	5645	**4352**	4428	2247	2433	**1189**	1213
AMF	*PR*	.0357	.0316	**.0164**	.0167	.0092	.0085	**.0048**	**.0048**
	RE	.3565	.3165	**.1644**	.1667	.0922	.0846	**.0482**	.0484
	nDCG	.2421	.2147	**.1010**	.1030	.0462	.0419	**.0228**	.0231
	EFD	.5987	.5184	**.2303**	.2352	.0971	.0853	**.0442**	.0447
	SE	9.98	8.90	**7.19**	7.20	8.30	7.41	**6.30**	6.30
	ICov	3847	2708	**2315**	2321	1486	1169	**1066**	1077

iterations. We fix the iteration number and study how `MSAP` impairs the RS varying the perturbation budget ϵ.

On the **perturbation side**, by looking at Figure 7.1a, one can note that both `MSAP` strategies are more powerful compared with the single-step one, for a fixed perturbation budget $\epsilon = 0.5$. For instance, the PGD perturbation technique shows **15.1** (0.1216 v.s. 0.0080), **20.4** (0.1216 v.s. 0.0060), and **23.8** (0.1216 v.s. 0.0051) times stronger impact with respect to FGSM, for iterations 25, 40, and 50 respectively. These results confirm CV's findings on the superiority of `MSAP`— in terms of model damage — compared to single-step methods in RSs. To better reveal `MSAP` effects, analyzing Figure 7.1a, we observe that after 25 iterations, the perturbed *BPR-MF starts to perform similar to the random recommender*. In other words, BPR-MF has lost all the learned users' personalized information.

Moreover, Table 7.1 confirms that `MSAP` strategies outperform FGSM for all <dataset, recommender> combinations. For instance, the <`ML-1M`, BPR-MF> combination shows the PGD perturbations reduced the accuracy by more than 2 times compared to FGSM, e.g., (0.0074 v.s. 0.0035), (0.0740 v.s. 0.0353), and (0.0368 v.s. 0.0172) for *PR*, *RE*, and nDCG, respectively. Here, we should point out that both Figure 7.1 and table 7.1 do not show a clear difference in PGD perturbation compared to BIM perturbation. This finding is different from the one previously reported in [33] for CV. We motivate it because tested model-based recommenders are less sensitive to the embedding initialization than the weight initialization of neural networks in the CV domain, since BPR computes gradients based on the differences between pairs.



(a) *nDCG* on BPR-MF

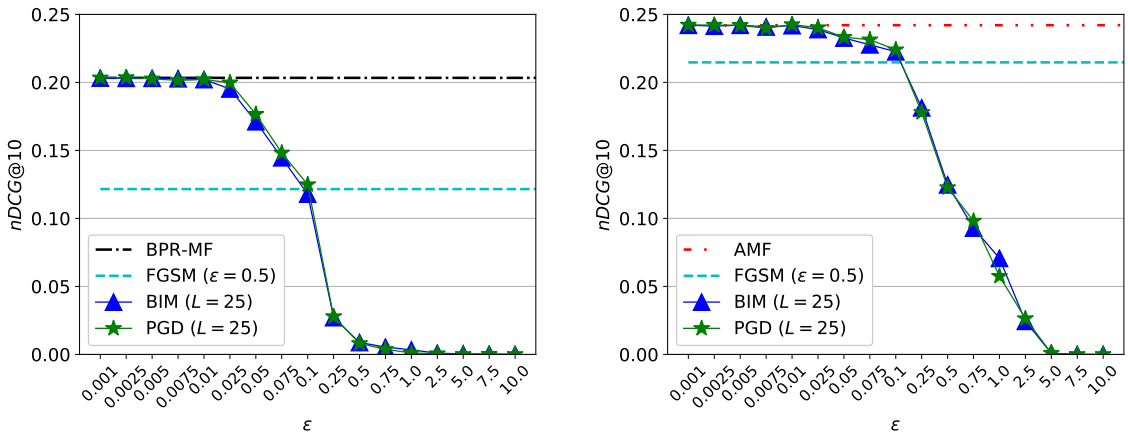
(b) *nDCG* on AMF

(c) *ICov* on BPR-MF

(d) *ICov* on AMF

Fig. 7.1 *nDCG* and *ICov* results for `LastFM`. Results of the (baseline) random recommender are in violet dotted line.

For what concerns beyond accuracy analysis, we found an interesting behavior for the BPR-MF. During the first 25 iterations of BIM, *ICov* increments nearly by 76% (from 6,220 to 10,928) compared to the coverage value of the non-perturbed recommender (see Figure 7.1c). After that, it steadily diminishes with a minimum *ICov* value of 1,948 (for BIM). This result, strengthened by looking into Table 7.1, may be justified because when `MSAP` computes several iterations ($L \geq 70$), it steadily destructs the accuracy metrics and brings the model to recommend a set of few items that all the users will not appreciate. Thus, we can conclude that `MSAP` deteriorates the personalized recommender to perform as bad as a random recommender (see Figure 7.1a) on both accuracy and beyond-accuracy measures.



(a) BPR-MF (varying ϵ) (b) AMF (varying ϵ)

Fig. 7.2 `MSAP` results varying $\epsilon \in [0.001, 10.0]$ on `LastFM` ($L = 25$). Figures 7.2a and 7.2b show that with a small perturbation, e.g., $\epsilon \simeq 0.1$, `MSAP` is more effective than FGSM with $\epsilon = 0.5$.

On the **defensive side**, Figure 7.1b shows an evident performance drop in accuracy for AMF which is, on average, more than 58% for `MSAP` and 11.31% for FGSM (see Table 7.1). For instance, the PGD perturbation shows **1.48** (0.2147 v.s. 0.1448), **1.86** (0.2147 v.s. 0.1154), and **1.94** (0.2147 v.s. 0.1106) times stronger impact with respect to FGSM, for iterations 20, 30, and 50, respectively. However, the accuracy reduction does not reach random performance as for the BPR-MF recommender. We may explain this slight robustness by mentioning the partial effectiveness of the adversarial regularization procedure, i.e., specifically designed to protect against FGSM [115].

Impact of `MSAP` varying ϵ.

In this study, we relax the investigation of the impact of iteration increase on iterative attacks' performances. Instead, by fixing the number of iterations (i.e., $L = 25$, the value previously shown to be the critical point (the elbow of the curve in Figure 7.1a) in performance deterioration) and varying ϵ from 0.001 to 10, we investigate at what ϵ-level, iterative attacks can get a similar performance comparable with FGSM. Analyzing Figures 7.2a and 7.2b, we found that iterative adversarial strategies reach the FGSM ($\epsilon = 0.5$) performance at iteration-level $\epsilon \simeq 0.1$. In other words, by using $^{0.5}/_{0.1}$ =**5** times less perturbation budget, the new iterative strategies reach a similar performance as that of the state-of-the-art FGSM attack strategy, independently of the recommender, i.e., the defense-free BPR-MF or the adversarial defended AMF.

In summary, the results of the two above studies provide strong evidence that:

- **Contribution 1** iterative attacks for the item recommendation task are more potent than the single-step FGSM strategy widely adopted in the prior literature of the RS community. For example, with only 25 iterations, the new attack strategies reduce the BPR-MF performance by an amount of 15 times (along nDCG) for a fixed perturbation budget $\epsilon = 0.5$, i.e., they are as effective as FGSM by using only 20% of the perturbation budget ($\epsilon \simeq 0.1$);
- **Contribution 2** the state-of-the-art defensive strategy explored in the RS community (i.e., APR) can diminish the impact of iterative attacks. However, these attacks still have a high capability to impact and impair the quality of the defended AMF recommender. These results suggest the need to identify mediating factors that can reduce the impact of iterative attacks against RS but are left for future investigation.

Investigating the `MSAP` Effects on the RS Fairness (RQ3)

This section analyses the impact of attacking a recommender system, i.e., BPR-MF, under a fairness perspective. Fairness analysis is becoming increasingly important in the last years in several machine learning-related fields. Recommendation algorithms are prone to generate algorithmic biases, reproduce biases in data, or acquire prejudices in training data [39, 246, 74]. In this scenario, analyzing fairness is more important than ever since a substantial variation of recommendation performance for the different groups of users, or categories of items, may unveil the attacker.

To this purpose, we have measured the accuracy performance considering the different groups/categories and three fairness metrics, namely GCE,MADR, andMADR, exploring both the initial and attacked models to capture the correct behavior and contrast it against the observed one after the attacks. In these experiments, we have evaluated BIM and PGD with 150 iterations, since at this point, the attack is very effective (low accuracy and beyond accuracy metrics). In detail, GCE considers several possible ideal probability distributions for each user, or item, clustering. Hence, it computes the divergence of the recommendation results (by considering a specific metric, i.e., nDCG) from the ideal distributions. Consequently, GCE with a value closer to zero denotes the recommender's congruence with that specific probability distribution. On the other hand,MAD focuses on the absolute variation of a given metric from an ideal situation in which groups/categories are treated equally. The original formulation ofMAD, namelyMADR, considers the <user, item>'s scores pairs in the recommendation results. Additionally, we have also considered a later extension

Table 7.2 Performance (measured in terms of nDCG) of the different approaches on each subset of users/items, where C_1 and C_4 denote the least and most popular items and users with less and more interactions, respectively; for user gender C_1 is associated to males and C_2 to females. Results for `ML-1M` are presented on the left, `LastFM` on the right. We highlight in bold the best results for each model.

Model		Item pop				User gender		User interactions			
		C_1	C_2	C_3	C_4	C_1	C_2	C_1	C_2	C_3	C_4
BPR-MF	initial	**0.054**	**0.035**	**0.045**	**0.300**	**0.046**	**0.043**	**0.079**	**0.044**	0.032	**0.023**
	FGSM	0.027	0.017	0.043	0.284	0.044	0.041	0.073	0.044	**0.032**	0.022
	BIM	0.005	0.000	0.000	0.167	0.019	0.016	0.018	0.020	0.018	0.016
	PGD	0.000	0.000	0.000	0.178	0.017	0.016	0.022	0.018	0.015	0.012
AMF	initial	**0.172**	0.096	0.096	**0.334**	**0.047**	**0.043**	**0.078**	**0.047**	**0.034**	**0.026**
	FGSM	0.163	**0.114**	**0.110**	0.326	0.043	0.039	0.070	0.041	0.033	0.022
	BIM	0.000	0.000	0.000	0.198	0.022	0.018	0.024	0.018	0.025	0.018
	PGD	0.002	0.055	0.000	0.202	0.023	0.017	0.024	0.018	0.025	0.018

Model		Item pop				User gender		User interactions			
		C_1	C_2	C_3	C_4	C_1	C_2	C_1	C_2	C_3	C_4
BPR-MF	initial	**0.000**	0.000	**0.006**	**0.092**	**0.218**	**0.143**	**0.158**	**0.209**	**0.194**	**0.253**
	FGSM	0.000	**0.001**	0.004	0.062	0.131	0.085	0.102	0.118	0.123	0.143
	BIM	0.000	0.000	0.000	0.004	0.007	0.009	0.011	0.007	0.009	0.002
	PGD	0.000	0.001	0.000	0.002	0.004	0.006	0.007	0.005	0.004	0.004
AMF	initial	0.000	**0.006**	**0.014**	**0.106**	**0.260**	**0.188**	**0.174**	**0.237**	**0.229**	**0.329**
	FGSM	0.000	0.000	0.010	0.095	0.230	0.168	0.153	0.211	0.198	0.297
	BIM	**0.002**	0.001	0.005	0.046	0.098	0.066	0.052	0.081	0.086	0.143
	PGD	0.000	0.002	0.003	0.046	0.097	0.061	0.054	0.082	0.090	0.142

ofMAD proposed in [74],MADR, in which the per-user performance values of an accuracy metric (i.e., nDCG) are considered.

Before focusing on fairness, let us analyze the behavior of recommenders for the different groups/categories to uncover the potential biases produced or removed by the attack strategies. Table 7.2 depicts the nDCG performance of the recommenders (BPR-MF, AMF, and their attacked variants) regarding the clusters for three attributes: item popularity, user gender, and user interactions. The clustering for item popularity and user interactions was computed by considering the quartiles for the attributes, while user gender is naturally clustered in the original datasets. This table shows, as already noted in the literature, BPR-MF achieves higher values of nDCG for popular items for both `ML-1M` and `LastFM`; in this respect, note the performance of BPR-MF in C_4 regarding the item pop attribute. Notably, the efficacy of the attacks is particularly evident here since, for BPR-MF, the C_4 for the item pop attribute column shows a degradation of the performance when the recommender is under attack.

Table 7.3 Fairness measured according to GCE where f_0 represents a uniform distribution, f_k denotes a distribution where the group C_k accumulates more probability than the rest, as in $f_1 = [0.75, 0.25]$ for user gender, MADr, and MADR. Rest of notation as in Table 7.2.

Data	Model		Item pop					User gender					User interactions				
			f_0	f_1	f_4	$MADr$	$MADR$	f_0	f_1	f_2	$MADr$	$MADR$	f_0	f_1	f_4	$MADr$	$MADR$
ML-1M	BPR-MF	initial	**−0.483**	**−1.574**	**−0.005**	0.040	0.159	−0.001	−0.109	−0.143	**0.050**	0.003	−0.116	**−0.138**	−1.480	0.618	0.030
		FGSM	−0.929	−3.056	−0.042	**0.029**	0.140	**0.000**	−0.111	**−0.140**	0.067	0.002	−0.110	−0.158	−1.514	**0.614**	0.028
		BIM	−2,039.764	−334.326	−326.189	0.066	**0.079**	−0.003	**−0.088**	−0.170	0.373	0.003	**−0.004**	−0.542	**−0.679**	1.781	**0.002**
		PGD	−3,167.250	−8,615.699	−506.580	0.062	0.083	0.000	−0.111	−0.140	0.234	**0.001**	−0.024	−0.323	−0.910	1.564	0.005
	AMF	initial	−0.147	**−0.576**	**−0.105**	0.225	0.424	−0.001	−0.104	−0.149	0.084	0.004	−0.092	**−0.162**	−1.329	1.995	0.028
		FGSM	**−0.104**	−0.646	−0.121	0.171	0.302	**−0.001**	−0.105	**−0.147**	**0.038**	**0.004**	−0.093	−0.166	−1.403	**1.674**	0.025
		BIM	−3,533.378	−9,611.568	−565.161	**0.095**	**0.155**	−0.007	−0.074	−0.193	0.302	0.005	−0.014	−0.435	−0.719	4.175	0.005
		PGD	−1,543.481	−287.878	−246.845	0.263	0.330	−0.011	**−0.064**	−0.213	0.328	0.006	**−0.010**	−0.426	**−0.702**	4.177	**0.004**
			f_0	f_1	f_4	$MADr$	$MADR$	f_0	f_1	f_2	$MADr$	$MADR$	f_0	f_1	f_4	$MADr$	$MADR$
LastFM	BPR-MF	initial	−1,161.806	−4,646.677	−185.725	0.120	0.032	−0.016	−0.188	−0.499	0.147	0.051	−0.015	−0.822	**−0.353**	**0.557**	0.051
		FGSM	−397.483	−3,094.772	−63.435	0.123	0.033	−0.016	**−0.180**	−0.489	**0.141**	0.031	**−0.008**	−0.730	−0.395	0.686	0.022
		BIM	−46.740	−186.212	−7.314	0.031	0.003	**−0.002**	−0.372	−0.258	0.312	**0.001**	−0.206	**−0.243**	−2.591	3.153	0.005
		PGD	**−20.224**	**−156.603**	**−3.149**	**0.021**	**0.002**	−0.011	−0.480	**−0.243**	0.290	0.001	−0.025	−0.282	−0.694	3.062	**0.002**
	AMF	initial	−747.062	−5,853.077	−119.395	**0.468**	0.055	−0.010	−0.190	−0.416	0.224	0.048	**−0.026**	**−0.921**	−0.291	2.057	0.079
		FGSM	−1,242.414	−4,969.776	−198.632	0.583	0.066	**−0.009**	−0.193	**−0.413**	**0.108**	0.042	−0.028	−0.930	−0.279	**1.060**	0.074
		BIM	**−2.238**	**−8.706**	**−0.217**	0.672	**0.035**	−0.014	**−0.178**	−0.459	0.941	**0.021**	−0.067	−1.257	**−0.200**	6.127	0.046
		PGD	−309.333	−2,419.092	−49.342	0.742	0.039	−0.022	−0.200	−0.562	0.978	0.025	−0.063	−1.237	−0.210	7.015	**0.046**

On the other hand, when the recommender is defended, i.e., AMF, the performance deterioration is less evident, even though the trend in the approaches remains the same. Considering the user gender, we observe that the recommendation performance for males (C_1) is higher than for women in both datasets. Even though the trends are similar to those observed for item popularity, it is worth noticing that the degradation and the defense effects are more evident in `LastFM`. Finally, the table shows two opposite behaviors for user interactions: in `ML-1M`, BPR-MF seems to favor the less active users, whereas `LastFM` favors the most active ones. The reason for this behavior is probably twofold. First, in `ML-1M`, there are no proper cold-users: the minimum number of interactions is 19, and there are $1,522$ users in C_1 with several interactions that range from 19 to 43. In `LastFM`, on the other hand, there are only 716 users in C_1, involving users with interactions from 2 to 123. Second, the datasets show a dramatically different number of items in the catalogs, thus making the number of interactions sufficient to produce meaningful recommendations for `ML-1M`.

Regarding the change in performance when using any of the attack methods, we observe that in `ML-1M` the trend and absolute values remain almost the same with respect to the initial recommender; however, in `LastFM` the situation is not identical: while the degradation follows the same trend, defended methods (AMF) show higher accuracy values for all the clusters. Once we have analyzed the performance found on an attribute basis (for some sensitive attributes), we show in Table 7.3 the result of the fairness-aware evaluation metrics described before. We first analyze which initial methods better approximate ideal distributions, and whether this situation changes when we use a defended model.

With this goal in mind, we analyze the GCE fairness values corresponding to the initial methods, without and with defense. We observe a consistent behavior in both datasets: the order derived from the GCE values is the same for BPR-MF and AMF. However, for some cases, the actual values are different, meaning that the defendant variant diverges differently (either more or less) from that distribution than the original method; for instance, for item popularity in `ML-1M`, the uniform (f_0) and least popular items (f_1) obtain a lower absolute GCE value for the defended model, whereas the behavior is the other way around for user interactions in `LastFM`. An interesting case is one of the user genders, wherein `ML-1M` the divergence for males (f_1) is decreased, whereas in the `LastFM` experiments, is the opposite; this evidences a non-predictable effect of the defended models with respect to some attributes.

Let us now study whether the defense and attack methods modify the fairness performance. For this, we observe that some attack methods like BIM help to increase the fairness on some distributions (or attribute values) at the expense of others, such as f_1 for user gender and f_4 for user interactions in `ML-1M`, at the expense of f_2 and f_1 respectively. Finally, we explore whether any attribute is more sensitive under a fairness perspective, since this may be a strong signal that a recommender is under attack. Thus, we note that FGSM tends to obtain very similar GCE values andMADRvalues in almost every scenario, whereasMADr tends to change whenever an attack is performed. Because of this, we conclude that if we measure fairness based on ranking performance (i.e., according to GCE orMADR), an FGSM attack might go unnoticed, whereas MADr is more sensitive to any attack. On the other hand, the rest of the attack strategies seem to change the distribution of the recommendations, as it becomes evident in the GCE values of item popularity.

7.4 Related Work

The research contributions of the current chapter have to be placed in the research line of adversarial machine-learned perturbation of model parameters presented in Section 2.3.2. We propose to study the application of AML techniques to generate perturbations to reduce recommenders' performance and their countermeasures [115, 37]. While the work [115] reported serious vulnerability of BPR-MF against adversarial perturbation obtained from the FGSM attack and suggested an adversarial regularization procedure as a defensive countermeasure. This chapter inspired other recommender models (and studies) such as AMR [209], FG-ACAE [234, 235], and ATF [58]. However, we have found that the RS community lacks studies on other categories of adversarial

perturbations such as iterative attacks (e.g., PGD [155]). Indeed, in the CV domain, iterative adversarial perturbations have been demonstrated to improve the attack effectiveness by more than 60% compared to FGSM [142]. However, to the best of our knowledge, no major attempt has been made in the RS community to study the RS performance variation when multi-step perturbations alter model embeddings. To fill this gap, in this chapter we have proposed `MSAP`, the first iterative perturbation method proposed to study the robustness/stability in the recommendation task.

7.5 Summary

In this chapter, we proposed iterative adversarial attacks against personalized recommenders models. We studied the impact of the proposed attacks with extensive experiments on two datasets (i.e., `LastFM` and `ML-1M`) and two state-of-the-art recommenders, i.e., BPR-MF and AMF — an extension to the BPR-MF that integrates the adversarial training as the defense against single-step attacks. Our experiments show that under a fixed perturbation budget, the presented multi-step attack strategies, namely the basic iterative method (BIM) and projected gradient descent (PGD), are considerably more effective than the state-of-the-art single-step FGSM method. We verified the degradation of recommendation quality along with accuracy, beyond-accuracy, and fairness metrics. In particular, experiment validations showed two main messages. The first is that non-defended recommenders perturbed by the multi-step attack strategies can be impaired/weakened so much that their performance becomes worse than a random recommender. The second claims that even the adversarially defended model against FGSM can lose half of its recommendation performance (i.e., after being confronted with an iterative attack, they preserve only half of the learned personalized users' preferences). Equivalently, we verified that iterative attacks could produce the same performance drop as FGSM attacks with 5-time smaller perturbation levels. These results evidence the vulnerability of the personalized BPR-learned models, both in defended and non-defended scenarios.

Additionally, we analyzed how attacks might produce variations on the fairness of a recommender model. By clustering the items by their popularity and users by their interactions and gender, we verified that, differently from single-step attacks, the presented multi-step strategies changed the fairness measurements considerably. We plan to investigate defense strategies against the analyzed iterative attacks. Moreover, we intend to extend the fairness evaluation by exploring other attribute-based clusters and novel methods.

Chapter 8

Theoretical Modeling of Adversarial Training on Recommendations

Since adversarial training has been demonstrated to disturb the model accuracy in the image classification task, how does it influence the recommendation performance on accuracy and beyond-accuracy perspectives?

RSs employ user-item feedback, e.g., ratings, purchases, or reviews, to match customers to personalized lists of products. Approaches to top-K recommendation mainly rely on Learning-To-Rank algorithms and, among them, the most widely adopted is Bayesian Personalized Ranking (BPR), which bases on a pairwise optimization approach. Recently, BPR has been found vulnerable against adversarial perturbations of its model parameters. Adversarial Personalized Ranking (APR) mitigates this issue by robustifying BPR via an adversarial training procedure. The empirical improvements of APR's accuracy performance on BPR have led to its wide use in several recommender models. However, a key overlooked aspect has been the beyond-accuracy performance of APR, i.e., novelty, coverage, and amplification of popularity bias, considering that recent results suggest that BPR, the building block of APR, is sensitive to the intensification of biases and reduction of recommendation novelty.

In this chapter, we model the learning characteristics of the BPR and APR optimization frameworks to give mathematical evidence that, when the feedback data have a tailed distribution, APR amplifies the popularity bias more than BPR due to an unbalanced number of received positive updates from short-head items. We empirically validate the theoretical results using matrix factorization (MF) by performing an extensive experimental study on five public datasets to compare BPR-MF and APR-MF performance on accuracy and beyond-accuracy metrics. The experimental results

consistently show the degradation of novelty and coverage measures and a worrying amplification of popularity bias.

8.1 Introduction

Machine-learned models such as latent factor models (LFMs) have significantly advanced the capability of recommender systems (RSs) to be faster and more accurate. Learning from historical users' preferences, i.e., ratings, purchases, or clicks, is essential to personalization and facilitating a better user experience. To address this task, modern RS often employ Bayesian Personalized Ranking (BPR) [188], a pairwise ranking optimization framework that uses item pairs as training data and optimizes it for correctly ranking item pairs. BPR is currently the state-of-the-art optimization framework for computing personalized ranking in RS and has been widely adopted in many research works [121, 61, 225].

Notwithstanding their great success, lately, it has been shown that ML applications can be *adversarial in nature* [216]. Recent works have shown the fragility of BPR-based trained recommender when confronted with *adversarial perturbations*, i.e., small but non-random perturbations added to the recommender model parameters, to cause recommendation performance [115]. Several works have shown the vulnerability of LFMs trained with BPR under adversarial attacks, for instance, He et al. [115] empirically verify that adversarial perturbation of BPR-MF, i.e., a matrix factorization (MF) model trained with BPR, decreases the nDCG metric value by -26.3%. Yuan et al. [235] show the same degradation trend with perturbations applied against the parameters of collaborative auto-encoder (CAE) models. Chen and Li [58] verify the weakness of the tensor factorization (TF) approach, and Tang et al. [209] validate the non-robustness of personalized visual-based recommenders (VBPR) under adversarial attacks.

To address this issue, as a defensive remedy, He et al. [115] propose Adversarial Personalized Ranking (APR), a novel optimization strategy to robustify BPR against adversarial perturbations. Based on the *adversarial training* procedure proposed by Goodfellow et al. [101], APR extends BPR by integrating the BPR-objective function with an additional regularization term, named *adversarial regularizer*, that quantifies the loss value when the model parameters are adversarially perturbed. The robustified version of BPR showed a nDCG reduction of only -2.9% on MF [115], a protection effect confirmed also on other models such as CAE [235], TF [58], and VBPR [209]. The key insight is that APR not only improves the defensive capability

Table 8.1 List of articles proposing novel recommendation algorithms employing APR as the optimization strategy.

Article	**Conference**	**Year**
He et al. [115]	SIGIR	2018
Yuan et al. [234]	IJCNN	2019
Yuan et al. [235]	SIGIR	2019
Tran et al. [211]	SIGIR	2019
Chen and Li [58]	RecSys	2019
Park and Chang [179]	WWW	2019
Dai et al. [71]	WWW	2019
Feng et al. [95]	TKDE	2019
Wang et al. [221]	IET	2019
Liu et al. [152]	IEEE ITAIC	2019
Manotumruksa et al. [156]	SIGIR	2020
Li et al. [148]	WSDM	2020
Yuan et al. [236]	WSDM	2020
Wang and Han [222]	IEEE Access	2020
Tang et al. [209]	IEEE TKDE	2020
Weibo et al. [226]	Applied Intelligence	2021

of RS (robustness under adversarial attacks) but also their generalization performance in normal item recommendation tasks. For instance, He et al. [115] show that for optimizing MF, if APR is used instead of BPR, a relative improvement of +11% on accuracy performance is achieved when compared to the results obtained with BPR.

Given the gained performances obtained in both robustness and accuracy dimensions, we have recently witnessed the application of APR in a growing number of research works. Table 8.1 presents a list of more than 15 articles where a novel recommendation algorithm has been proposed incorporating the APR as the core optimization framework. These examples underline the popularity of the adversarial ranking-based procedure, i.e., APR, for various item recommendation tasks. However, given the sensitivity of BPR against popularity bias reported in recent works [130, 1, 247, 46], the question remains as to how much APR is vulnerable against popularity bias and its amplification considering that BPR is the APR building block.

Motivated by this observation, the chapter at hand focuses its attention on the learning differences between APR and BPR to understand how much beyond-accuracy measures, including novelty, coverage, and influence of tailed data distributions, could be affected by APR. We formally study the learning characteristics of both optimization strategies to quantify the consequences of the adversarial regularization procedure. The

proposed analysis is supported by an extensive empirical evaluation of the performance variations produced when using APR in recommendation datasets with data-tailed distributions.

The main contributions presented here include:

- the presentation of a formal analysis to identify whether APR is affected by popularity amplification bias, and highlighting how difference such bias is in comparison with BPR —the core building block used in APR;
- the empirical verification of the existence of a trade-off between accuracy and beyond-accuracy measures and popularity bias in APR;
- the study on the accuracy and beyond-accuracy performances when varying two APR hyper-parameters: the adversarial perturbation budget (ϵ) and the adversarial regularization coefficient (α).

An experimental evaluation has been carried out on five recommendation datasets using MF as the base ML model. The results motivate the design of novel pairwise robust learning procedures that can strike a more meaningful balance between accuracy, beyond accuracy, and low amplification of popularity bias.

8.2 Formal Analysis

In this section, we formally define the recommendation task as a learning-to-rank problem. Then, we introduce BPR and APR optimization techniques before moving to the definition and comparison of both approaches' gradient magnitudes. In the end, we formally identify that a possible phenomenon of amplification of the item-popularity bias could affect APR-based model performance.

8.2.1 Recommender System Formalization

The item recommendation task builds a user's personalized list of K items ranked by predicted relevance scores. Given a user $u \in \mathcal{U}$, the rank of a not-interacted item $i \in \mathcal{I}$ is defined via the bijective function in $\mathcal{I}$ as $\hat{r}(i|u)$. The ranking function $\hat{r}(\cdot)$ is based on the predicted value of the preference score function $\hat{s}(\cdot|\Theta)$. Θ represents the ML recommender's model parameters, e.g., matrix factorization (MF) [139]. To build the top-K recommendation list associated with the user u, the user's not-interacted items

are sorted in decreasing order by the predicted preference score. Formally, the rank of each item is defined as

$$\hat{r}(i \mid u) := \left\{ |\{j : \hat{s}(j \mid u) \geq \hat{s}(i \mid u)\}|, i, j \in \mathcal{I} \backslash \mathcal{I}_u^+ \right\} \tag{8.1}$$

where $\mathcal{I}_u^+$ is the list of (positive) items already seen by the user u.

8.2.2 Bayesian Personalized Ranking

The model parameters Θ can be learned with optimization procedures. The three most implemented approaches are point-wise [139], pair-wise [188], and list-wise [199]. Among them, the pair-wise learning with BPR is a standard strategy in several state-of-the-art recommender models, e.g., recurrent neural models [121], attentive collaborative recommenders [61], or neural graph learning [225].

BPR assumes that given a user u, the score $\hat{s}(i|u)$ predicted on an already interacted item $i \in \mathcal{I}_u^+$ should be higher than the one estimated for a not-interacted item $j \in \mathcal{I} \backslash \mathcal{I}_u^+$. Commonly, the first item is called *positive*, while the seconds *negative*. A user u, a positive item i, and a negative item j form (u, i, j) a training triplet. It follows that, the full set of pair-wise preferences $\mathcal{D}_\mathcal{R} \subseteq \mathcal{U} \times \mathcal{I} \times \mathcal{I}$ is composed by all the triplets (u, i, j) such that:

$$(u, i, j) \in \mathcal{D}_\mathcal{R} :\Longleftrightarrow \left(i \in \mathcal{I}_u^+ \wedge j \in \mathcal{I} \backslash \mathcal{I}_u^+ \right) \tag{8.2}$$

To build $\mathcal{D}_\mathcal{R}$, it is necessary to define the sampling strategy of negative items. BPR associates a negative item j to each (u, i)-pair by uniformly sampling j from the set of u not-interacted ones $(\mathcal{I} \backslash \mathcal{I}_u^+)$. Since BPR associates a single negative item to each recorded pair of interactions, it follows that the size of $\mathcal{D}_\mathcal{R}$ is equal to the number of recorded preferences. Consequently, the dimension of $\mathcal{D}_\mathcal{R}$ is smaller than the number of all possible interactions, i.e., R.

To learn Θ via BPR, Rendle et al. [188] define the optimization problem

$$\underset{\Theta}{\operatorname{argmax}} \prod_{(u,i,j) \in \mathcal{D}_\mathcal{R}} \sigma(\hat{s}(i|u) - \hat{s}(j|u)) \tag{8.3}$$

where $\sigma(\cdot)$ is the sigmoid function, i.e., $\sigma(z) = 1/(1 + e^{-z})$. The maximization problem defined in Equation (8.3) can be equivalently solved by minimizing the negative log-likelihood

$$\underset{\Theta}{\operatorname{argmin}} \underbrace{- \sum_{(u,i,j) \in \mathcal{D}_\mathcal{R}} \ln \sigma(\hat{s}(i|u) - \hat{s}(j|u))}_{:= \mathcal{L}_{BPR}} \tag{8.4}$$

where, the $\mathcal{L}_{BPR}$ indicates the BPR objective function.

The standard technique to learn Θ is the stochastic gradient descent (SGD). Given a triplet $(u,i,j) \in \mathcal{D}_{\mathcal{R}}$, the model parameters are updated as defined below.

$$\Theta \leftarrow \Theta - \eta \frac{\partial \mathcal{L}_{BPR}(\Theta)}{\partial \theta} \tag{8.5}$$

$$\Theta \leftarrow \Theta + \eta(1 - \sigma(\hat{s}_{uij}(\Theta))) \frac{\partial \hat{s}_{uij}(\Theta)}{\partial \Theta} \tag{8.6}$$

where η is the learning rate. In the following, we will use $\hat{s}_{uij}(\Theta)$ to indicate the $\hat{s}(i|u) - \hat{s}(j|u)$ for lightening the formalism.

8.2.3 Adversarial Personalized Ranking

As we already said before, the BPR learned parameters are not robust to adversarial perturbations, as verified on several recommender models, e.g., matrix factorization [115], collaborative auto-encoders [235, 234], visual-based recommender [209], tensor-factorization [58], collaborative neural models [148], sequential recommendations [156], and attentive song recommenders Tran et al. [211]. Adversarial personalize ranking (APR) is the state-of-the-art defensive technique proposed by He et al. [115] to stabilize the BPR learning of model parameters and make it robust to adversarial perturbations.

Adversarial Perturbation

Before reporting APR, it is necessary to describe how to compute an adversarial perturbation. The adversarial perturbation Δ_{adv} is

$$\Delta_{adv} := \underset{\Delta, ||\Delta|| \leq \epsilon}{\operatorname{argmax}} \mathcal{L}_{BPR}(\hat{\Theta} + \Delta) \tag{8.7}$$

where ϵ is the perturbation budget to limit the maximum amount of noise added to the Θ, $||\cdot||$ is the L_2-norm, and $\hat{\Theta}$ denotes the fixed model parameters on which the perturbation is evaluated. The intuition is that building a perturbation that increases the model's loss reduces the recommendation performance. Inspired by the fast gradient sign method by Goodfellow et al. [101], He et al. [115] solved Equation (8.7) by linearizing the objective function $\mathcal{L}_{BPR}$ as

$$\Delta_{adv} = \epsilon \cdot \frac{\Gamma}{\|\Gamma\|} \quad \text{where} \quad \Gamma = \frac{\partial \mathcal{L}_{BPR}(\hat{\Theta} + \Delta)}{\partial \Delta} \tag{8.8}$$

Adversarial Training

To robustify, and stabilize, the BPR-learned model against the performance drop caused by the adversarial perturbation, He et al. [115] proposed to use an *adversarial training* procedure. The procedure, named adversarial personalized ranking (APR), learns Θ within a minimax optimization game

$$\arg\min_{\Theta} \max_{\Delta_{adv}, \|\Delta_{adv}\| \leq \epsilon} \underbrace{\mathcal{L}_{BPR}(\Theta) + \alpha \mathcal{L}_{BPR}(\Theta + \Delta_{adv})}_{:= \mathcal{L}_{APR}(\Theta)} \tag{8.9}$$

where $\mathcal{L}_{APR}(\Theta)$, the APR objective function, is composed by the standard BPR loss, i.e., $\mathcal{L}_{BPR}$, and a regularization term, i.e., $\mathcal{L}_{BPR}(\Theta + \Delta_{adv})$, whose strength is controlled by α, named *adversarial regularization coefficient*. This additional regularization term, named adversarial regularizer, is the loss obtained when an adversarial perturbation Δ_{adv} is added to Θ to **maximize** the model objective (see Equation (8.7)). It follows that, being Δ_{adv} fixed, APR **minimizes** both the standard BPR loss $\mathcal{L}_{BPR}$ with, and without, Δ_{adv}. The aim of APR is to learn a model that is able to correctly distinguish the positive and negative items also in the case of adversarial perturbations.

As suggested in [115], the updates of Θ with APR are computed with SGD as follows:

$$\Theta \leftarrow \Theta + \eta \Big[(1 - \sigma(\hat{s}_{uij}(\Theta))) \frac{\partial \hat{s}_{uij}(\Theta)}{\partial \Theta} + \alpha (1 - \sigma(\hat{s}_{uij}(\Theta + \Delta_{adv}))) \frac{\partial \hat{s}_{uij}(\Theta + \Delta_{adv})}{\partial \Theta} \Big] \tag{8.10}$$

8.2.4 Gradient Magnitudes

Learning Θ with either BPR and APR is performed by looping over Equations (8.6) and (8.10), respectively. We present an approach to studying the learning of a recommender model, evaluating and comparing the updates' magnitudes of both pairwise optimizations.

The Bayesian Gradient Magnitude

Θ updates in Equation (8.6) depend on the learning rate η, the partial derivative of the difference of predicted scores $\hat{s}_{uij}(\Theta)$, and a multiplicative scalar $(1 - \sigma(\hat{s}_{uij}(\Theta)))$. Following Rendle and Freudenthaler [187], we define the *Bayesian gradient magnitude* (ω) with respect to the (u, i, j) triplet as

$$\omega_{uij} := (1 - \sigma(\hat{s}_{uij}(\Theta))) \tag{8.11}$$

This multiplicative scalar indicates how much the current model represented by Θ is performing in recognizing that the user u prefers the positive item i more than the negative item j.

The update of SGD significantly changes Θ when

$$\omega_{uij} \simeq 1 \implies \Big(\sigma(\hat{s}_{uij}(\Theta)) \simeq 0 \iff \hat{s}_{ui}(\Theta) \ll \hat{s}_{uj}(\Theta)\Big) \tag{8.12}$$

In this circumstance, the preference score $\hat{s}_{uj}$ predicted for the negative item j is bigger than the one predicted on the positive $\hat{s}_{ui}$. It follows that Θ requires a vast update within the current gradient step to learn how to correctly rank the items' in the (u, i, j) triplet. Conversely,

$$\omega_{uij} \simeq 0 \implies \Big(\sigma(\hat{s}_{uij}(\Theta)) \simeq 1 \iff \hat{s}_{ui}(\Theta) \gg \hat{s}_{uj}(\Theta)\Big) \tag{8.13}$$

is the scenario where the model does not need to update the parameters on the (u, i, j)-triplet since it has already learned how to recognize that u prefers i more than j.

The Adversarial Gradient Magnitude

Equation (8.10) extends Equation (8.6) with the addition of the adversarial regularization component. As stated in [115], APR is activated when BPR training is converging to robustify and stabilize the learning of Θ. Analyzing Equation (8.10), each APR gradient step has two multiplicative scalars: the already presented *Bayesian gradient magnitude* (ω), and another novel scalar, that we name *adversarial gradient magnitude* (ω^{adv}). Formally, ω^{adv} on (u, i, j) is defined as:

$$\omega^{adv}_{uij} := (1 - \sigma(\hat{s}_{uij}(\Theta + \Delta_{adv}))) \tag{8.14}$$

This quantity depends on how much the preference scores inferred from the perturbed model $(\Theta + \Delta_{adv})$ would be able to detect that u favors i more than j. It follows that, the ω^{adv}_{uij} value depends on the adversarial noise Δ_{adv} capability to revert the order preferences estimated by Θ. The adversarial case in which Θ necessitates a huge update to robustify the recommender model is

$$\omega^{adv}_{uij} \simeq 1 \implies \Big(\sigma(\hat{s}_{uij}(\Theta + \Delta_{adv})) \simeq 0 \iff \hat{s}_{ui}(\Theta + \Delta_{adv}) \ll \hat{s}_{uj}(\Theta + \Delta_{adv})\Big) \tag{8.15}$$

The previous case denotes the *worst-case* scenario when the model is not robust to the adversarial perturbation. Consequently, in the following *best-case* scenario,

$$\omega_{uij}^{adv} \simeq 0 \implies \Big(\sigma(\hat{s}_{uij}(\Theta+\Delta_{adv})) \simeq 1 \iff \hat{s}_{ui}(\Theta+\Delta_{adv}) \gg \hat{s}_{uj}(\Theta+\Delta_{adv})\Big) \quad (8.16)$$

the model does not require vast updates since the original user's preferences order is preserved in spite of the perturbations. Note that both ω_{uij} and ω_{uij}^{adv} depend on the model parameters (Θ) and thus they change for each gradient step.

8.2.5 Empirical Analysis of Gradient Magnitudes

As presented before, BPR and APR use SGD to update Θ. Figure 8.1 shows the probability of the Bayesian gradient magnitude (ω) and the adversarial gradient magnitude (ω^{adv}) measured during the training performed on two of the examined datasets, i.e., `Amazon` [159] and `ML100K` [109]. Figures 8.1a and 8.1b represent $p(\omega)$ measured for the BPR training with a number of training epochs $t \in [1,2,...,T_{BPR}]$ where $T_{BPR} = 100$, and both $p(\omega)$ and $p(\omega^{adv})$ when $t \in (T_{BPR}, T_{BPR}+1,...,T_{APR}]$ with $T_{APR} = 200$. The vertical red line in Figure 8.1 divides the probability measured with the initial BPR training with the ones measured when APR is activated after the T_{BPR}-epoch. [1].

Aligned with Rendle and Freudenthaler [187] empirical findings, Figures 8.1a and 8.1b show that after few training epochs ω is smaller than 0.01 for more than 85% of the training triplets of the `Amazon` dataset, and 65% for the `ML100K` ones. Next, we observe that the magnitudes measured on all the triplets is smaller than 0.5, i.e., $p(\omega_{uij} < 0.5) \simeq 1.0, \forall (u,i,j) \in D_{\mathcal{R}}$, after the first 50 epochs for both the datasets. The reduction of the Bayesian gradient magnitudes to values close to 0 after the first few training epochs is an already identified BPR *gradient vanishing* issue that leads to a slow model convergence [187].

Analyzing the behavior of the adversarial gradient magnitudes in Figures 8.1a and 8.1b, it can be observed that APR is not affected by the BPR gradient vanishing issues. For the `ML100K` dataset, it can be observed that all the APR lines (dotted curves) are lower than the BPR ones (continue curves), meaning that APR magnitudes are consistently higher than the BPR ones. This phenomenon is evident in the experiments on the `Amazon` dataset. Indeed, Figure 8.1a shows that the probability of getting small magnitudes, i.e., $p(\omega^{adv} < 0.1)$, is smaller than 10% also when 100 APR-training epochs

[1] All the empirical results presented in Chapter 8 refer to the matrix factorization (MF) model. Note that the analysis is reproducible with other models. We present additional MF details in Section 2.1.1

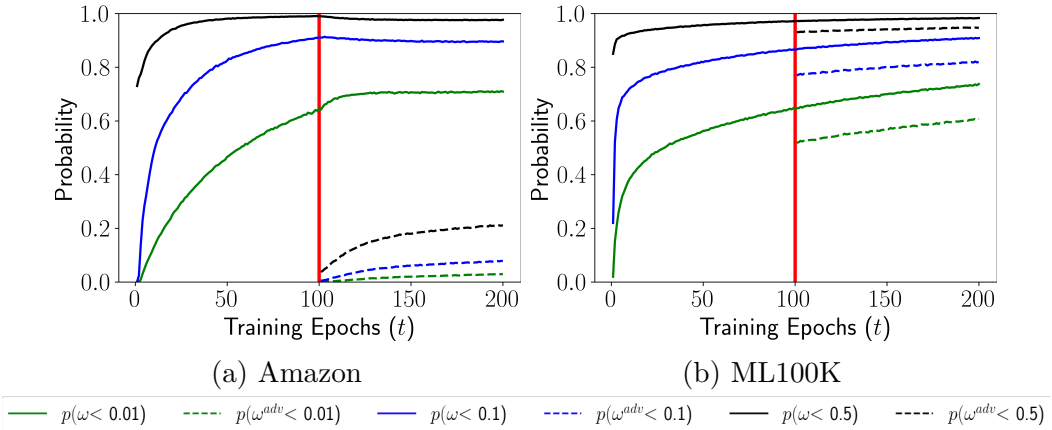


Fig. 8.1 Plots on the probability that a (u, i, j) triplet in $\mathcal{D}_{\mathcal{R}}$ has gradient magnitudes $\leq \{0.01, 0.1, 0.5\}$.

have been performed on the model. The reason is that the APR objective function also considers the adversarial regularizer. This regularizer forces further Θ updates to limit the performance drop due to an adversarial perturbation.

These results confirm that APR is a solution to both robustify and stabilize the BPR model training, as also claimed in [115]. Indeed, several works [115, 235, 209, 58] verified both a reduction of the adversarial perturbation efficacy in altering the recommendation performance and an increase of the accuracy measures when APR is employed to train the ML recommender model.

Before we move into an extensive empirical of the beyond-accuracy performance and popularity bias influences of APR, we study the impact of the imbalanced data distribution on APR learning in the next section.

8.2.6 Amplification of Popularity Bias

RS performance depends on structural and distributional characteristics of the user-item historical data [5]. Tailed data distribution is a property that received strong attention in the literature of RSs. Indeed, it is common in RSs that few items, named *short-head* items ($\mathcal{I}_{SH}$), receive much more feedbacks than many other ones, named *long-tail* ($\mathcal{I}_{LT}$) [1, 46]. In this work, we use the short-head and long-tail definition used by Abdollahpouri et al. [1], where the short-head set, composed of the top 20% of items by popularity, has much more feedback than the long-tail one, which contains the remaining 80% of items. Analyzing the datasets' statistics reported in Table 8.2, we can observe that the probability that positive feedback is a short-head item, i.e.,

$p(i|\mathcal{I}_{SH})$, is always more conspicuous than the probability of being in the long-tail set, i.e., $p(i|\mathcal{I}_{LT})$.

The primary motivation behind the study of tailed distributions' impact is the *amplification of popularity bias*. This means that a recommender model trained on non-uniformly distributed data could suggest popular (short-head) items more than niche (long-tail) ones, even when the latter would be of user's interest [206, 1, 2]. This phenomenon is also confirmed in BPR [130, 247, 46]. We conjecture that it could be important to understand whether APR could be affected, or even intensify, the amplification bias considering that APR hugely influences the BPR pre-trained model, as empirically verified in Section 8.2.5.

Effects of Imbalanced Data

Since the users' feedback data distribution is affected by popularity bias, the sampling of positive items follows the next relation

$$p(i \in \mathcal{I}_{SH}|u) \geq p(i \in \mathcal{I}_{LT}|u) \tag{8.17}$$

It means that the probability that a positive item of one triplet in $\mathcal{D}_{\mathcal{R}}$ is in the set of short-head items is higher than the probability of being in the long-tail (see Table 8.2). It follows that the uniform sampling of negative items used in both BPR and APR training strategies results in the relation

$$p(j \in \mathcal{I}_{SH}|u) = p(j \in \mathcal{I}_{LT}|u) = \frac{1}{|I|} \tag{8.18}$$

It means that the probability that the negative item in the (u,i,j)-training triple does not depend on the feedback distributions since they are randomly extracted from the complete set of items, i.e., I. From Equations (8.17) and (8.18) we deduce that the difference between the sampling distribution of positive and negative items to build $\mathcal{D}_{\mathcal{R}}$ could influence both the number and the sign of the model parameter updates made by BPR and APR optimization frameworks.

Theoretical Analysis

To formally study whether APR is affected by the amplification of popularity bias, we define two quantities: the *global positive* and *global negative* updates.

Definition 40 (**Global Positive Update** (Ω^+))**.** *Let* $t \in \{1,2,.,T_{BPR},T_{BPR}+1,..,T_{APR}\}$ *be a training epoch and* $\mathcal{D}_{\mathcal{R}}(t)$ *be the set of training triplets built for the* t*-th epoch,*

then the global positive update on short-head items is

$$\Omega^{+}(\mathcal{I}_{SH}|\mathcal{D}_{\mathcal{R}}(t)) := \sum_{(u,i,j)\in\mathcal{D}_{\mathcal{R}}(t)\wedge i\in\mathcal{I}_{SH}} \omega_{uij}(t) + \omega_{uij}^{adv}(t) \tag{8.19}$$

while the global positive update for long-tail items is

$$\Omega^{+}(\mathcal{I}_{LT}|\mathcal{D}_{\mathcal{R}}(t)) = \sum_{(u,i,j)\in\mathcal{D}_{\mathcal{R}}(t)\wedge i\in\mathcal{I}_{LT}} \omega_{uij}(t) + \omega_{uij}^{adv}(t) \tag{8.20}$$

Definition 41 (Global Negative Update (Ω^{-})). *The global negative update for short-head items at t-th training epoch is defined as follows*

$$\Omega^{-}(\mathcal{I}_{SH}|\mathcal{D}_{\mathcal{R}}(t)) := -\sum_{(u,i,j)\in\mathcal{D}_{\mathcal{R}}(t)\wedge j\in\mathcal{I}_{SH}} \omega_{uij}(t) + \omega_{uij}^{adv}(t) \tag{8.21}$$

while the global negative update for long-tail ones is

$$\Omega^{-}(\mathcal{I}_{LT}|\mathcal{D}_{\mathcal{R}}(t)) := -\sum_{(u,i,j)\in\mathcal{D}_{\mathcal{R}}(t)\wedge j\in\mathcal{I}_{LT}} \omega_{uij}(t) + \omega_{uij}^{adv}(t) \tag{8.22}$$

While Ω^{+} focuses on positive items (i), Ω^{-} focuses on negative ones (j). It follows that, using the inequality relations defined in Equations (8.17) and (8.18), we can derive that

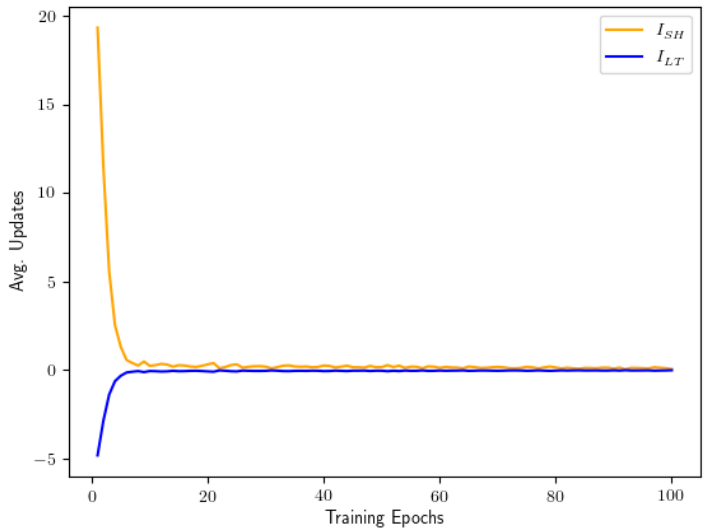
$$\Omega^{+}(\mathcal{I}_{SH}|\mathcal{D}_{\mathcal{R}}(t)) + \Omega^{-}(\mathcal{I}_{SH}|\mathcal{D}_{\mathcal{R}}(t)) \geq \Omega^{+}(\mathcal{I}_{LT}|\mathcal{D}_{\mathcal{R}}(t)) + \Omega^{-}(\mathcal{I}_{LT}|\mathcal{D}_{\mathcal{R}}(t)) \tag{8.23}$$

It implies that the global number of positive updates on short-head items is higher than the one on long-tail ones when a uniform distribution is used to sample the negative items and the users' feedback distribution is affected by popularity bias. It means that APR could be algorithmically affected by the **amplification of the popularity bias** as already checked on BPR. It is now necessary to verify whether APR amplifies, even more, the BPR issue.

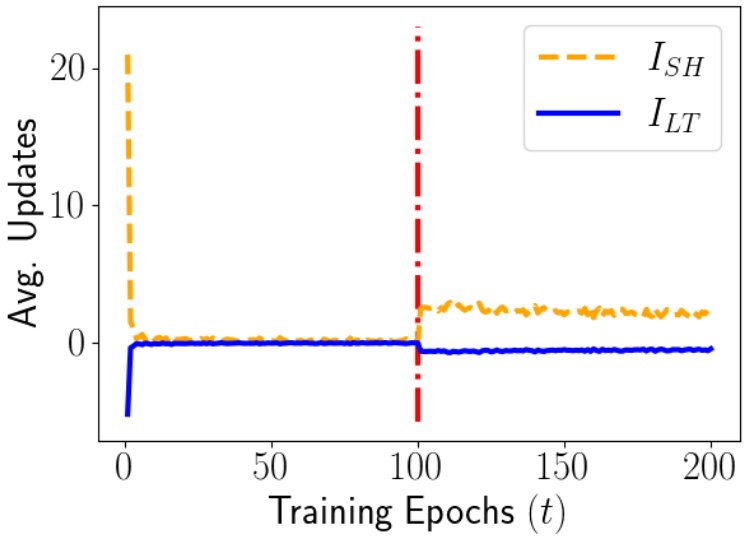
Empirical Validation: the Wine-Glass Phenomenon

Figures 8.2a and 8.2b show the $\Omega^{+}(\mathcal{I}_{SH}|\mathcal{D}_{\mathcal{R}}(t)) + \Omega^{-}(\mathcal{I}_{SH}|\mathcal{D}_{\mathcal{R}}(t))$ and $\Omega^{+}(\mathcal{I}_{LT}|\mathcal{D}_{\mathcal{R}}(t))$ $+\Omega^{-}(\mathcal{I}_{LT}|\mathcal{D}_{\mathcal{R}}(t))$ averaged by number of items in $\mathcal{I}_{SH}$ and $\mathcal{I}_{LT}$, respectively [2]. The first observation is that the sum of the first quantity is always positive for short-head

[2]The model configuration used in Figure 8.2 correspond to the best one shown in Table 8.3.



(a) Amazon



(b) ML100K

Fig. 8.2 Plots of the global gradient updates averaged by the number of items in $\mathcal{I}_{SH}$ and $\mathcal{I}_{LT}$. The red line indicates the start of APR.

items, while the second is negative for long-tail ones. Then, we identify a **wine-glass** phenomenon on the graphical representations of Figures 8.2a and 8.2b. In fact, each plot can be divided into three parts: the *base*, the *stem*, and the *bowl*. The *base* represents the BPR training epochs in which the updates on $\mathcal{I}_{SH}$ and $\mathcal{I}_{LT}$ have

Table 8.2 The statistics of the datasets.

Dataset	$\|\mathcal{U}\|$	$\|\mathcal{I}\|$	$\|\mathcal{R}\|$	Density	$p(i\|\mathcal{I}_{SH})$	$p(i\|\mathcal{I}_{LT})$
`ML100K`	943	1,682	99,999	0.0630	0.6452	0.3548
`Last.fm`	1,892	17,632	92,834	0.0028	0.7893	0.2107
`Amazon`	3,915	2,549	77,328	0.0077	0.5747	0.4253
`ML1M`	6,040	3,706	1,000,209	0.0447	0.6512	0.3488
`Yelp`	25,677	25,815	731,671	0.0011	0.6544	0.3456

an absolute magnitude different from 0. Already in this training phase, it can be seen that the average gradient magnitudes associated with $\mathcal{I}_{SH}$ are bigger than the one on $\mathcal{I}_{LT}$. This behavior is consistent with the well-known BPR amplification of popularity bias [130, 247, 157, 46]. The *stem* is the second component of the wine-glass. Observing Figures 8.2a and 8.2b, the last epochs of BPR ($T_{BPR}/2 < t \leq T_{BPR}$) show the gradient vanishing problem as examined in Section 8.2.5. In this phase, there is no amplification of bias since the model performs very tiny gradient updates. The last part of the glass, the *bowl*, exposes the average magnitudes in the case of APR training ($t > T_{BPR}$). It can be noted that the average sum of Bayesian and adversarial gradient magnitudes on each item in $\mathcal{I}_{SH}$ is much more notable than the one for $\mathcal{I}_{LT}$. These results empirically confirm Equation (8.23) and show that APR could increase even more than BPR the item popularity bias. We extensively examine the APR performance on beyond-accuracy and popularity bias metrics in the remainder of this work.

8.3 Experiments

Here, we present experimental settings and the discussion of the empirical results.

8.3.1 Settings

In this section, we introduce the datasets, evaluation measures, and evaluation protocol.

Datasets

We perform our experiments on five public datasets.

- **MovieLens 100K** (`ML100K`) [109] is a popular dataset with about 100,000 movie ratings popularly used for initial recommender model prototyping. We treat each rating as single positive feedback, indicating that a user likes the interacted film more than a not-interacted one.

- **Last.fm** [53] includes social networking, tagging, and music artist listening data from a set of 1,892 users of the `Last.fm` online music platform. We use the dataset version containing the list of all the artists listened to by each user. We model the artists as the items and the recorded listening as the feedback that a user prefers an artist.

- **Amazon** [159] holds product reviews given by the customers to the products on the e-commerce platform. As positive feedback, we utilize the user's purchases on the 'Grocery' vs. 'Tool' category. In particular, we use the dataset version released by Zhu et al. [247].

- **MovieLens 1M** (`ML1M`) [109] is an extended version of the `ML100K` movie dataset with a 10 times higher number of ratings (about 1 million).

- **Yelp** is a business review dataset released for the `Yelp` Challenge. We examine each user's review as a signal of interest toward business activity in the portal. We test the dataset version released by He et al. [115].

For each dataset, we employ the *leave-one-out* protocol [188, 115], putting in the test set either the last historical interaction when it is available, i.e., `ML100K`, `Amazon`, `ML1M`, and `Yelp`, or a random one, i.e., `Last.fm`.

Evaluation Metrics

We perform our analysis with the following set of measures.

- **Accuracy.** To study the accuracy performance, we report the precision ($Prec@K$), recall ($Rec@K$), and normalized discounted cumulative gain ($nDCG@K$) evaluated on the top-K recommendations [168].

- **Beyond-accuracy.** To measure the beyond-accuracy performance, we use the item coverage ($\text{Cov}_{\%}@K$) and the novelty (Nov). $\text{Cov}_{\%}@K$ measures the percentage fraction of the number of different items in the top-K recommendation lists on the size of the catalog (I). Values close to 100% indicates that the recommender model can generate recommendation lists covering almost the

entire catalog. Nov@K is defined as the capacity of the RS to generate novel and unexpected recommendations. We use the Nov@K metric proposed by Zhou et al. [243]. Following [243], given an item i, let $|\mathcal{U}_i|$ be the number of users who have previously interacted with i, let $|U_i|/|U|$ be the probability that a randomly selected user u has interacted with i, then the self-information associated to i is defined as $\mathrm{SI}_i = \log_2(u/|\mathcal{U}_i|)$. Let $\mathrm{MSI}_u@K$ be the mean self-information measured as the average SI associated to each item in the top-K recommendation list of u, then the novelty is $\mathrm{Nov}@K := \sum_{u\in\mathcal{U}} \mathrm{MSI}@K/|\mathcal{U}|$. Higher Nov@$K$ means a better RS ability to suggest unexpected items.

- **Popularity Bias.** To assess whether the recommendation lists are affected by the popularity bias, we adopt two sets of measures: (i) the long-tail diversity [2], and (ii) the ranking-based statistical metrics [247]. The first set includes the following metrics: the average recommendation popularity (ARP@K), the percentage of long-tail items (APLT@K), and the average coverage of long-tail items (ACLT@K). ARP@K evaluates the average popularity of the recommended items, APLT@K calculates the average fraction of long-tail items in each users' recommendation list, and ACLT@K measures the portion of recommended long-tail items. The second set includes the ranking-based statistical parity (RSP@K) and the ranking-based equal opportunity (REO@K). RSP@K measures the ratio between the recommendation probabilities for short-head ($P_{SH}@K$) and long-tail ($P_{LT}@K$) items. REO@K quantifies the previous recommendation probabilities considering the influence of the user's set of previously interacted items, i.e., $\hat{P}_{SH}@K$ and $\hat{P}_{LT}@K$. We refer to the original work by Zhu et al. [247] for further details.

Evaluation Protocol

We implemented the experimental framework using Tensorflow2. We fixed the size of the latent factor f to 64 as suggested in [115]. We trained the BPR-MF model for T_{BPR} epochs by varying the learning rate $\eta \in \{0.005, 0.01, 0.05\}$. After selecting the η hyper-parameters with the most accurate top-50 recommendations — accuracy measured as the recall (Re@50)— on the test set, we started the APR-MF training on the pre-trained BPR-MF model. We grid-searched the following set of APR-MF hyper-parameters: the perturbation budget $\epsilon \in \{0.001, 0.01, 0.1, 1.0\}$ and $\alpha \in \{0.001, 0.01, 0.1, 1.0, 10.0\}$. The APR-MF training is performed from the $T_{BPR}+1$ epoch, and it is completed until the T_{APR}-th epoch. We set $T_{BPR} = 100$ and $T_{APR} = 200$ for the smaller datasets,

i.e., `Amazon`, `Last.fm`, and `ML100K`; and $T_{BPR} = 1000$ and $T_{APR} = 1500$ for the last two bigger datasets. Note that we have also trained the BPR-MF model until the T_{APR}-th epoch to be fair in comparing the results of the BPR, and APR, MF models reported in the section 8.3.2. Further reproducibility details, the code, and the data are available on the public GitHub repository.

8.3.2 Results

This section presents the results and discusses the APR impact on accuracy, beyond-accuracy, and popularity objectives. Here, we aim to answer the following research questions:

RQ1 When APR improves the model accuracy, what are the effects on the beyond accuracy measures?

RQ2 Are the recommendation lists more affected by the popularity bias than those produced by BPR?

RQ3 How do the α and ϵ hyper-parameters affect the accuracy and beyond-accuracy of APR performance?

We report all the metric values on top-50 recommendation lists. For instance, we indicate Nov@50 as Nov, Re@50 as Re, nDCG@50 as nDCG. Additionally, we indicate with **R.V.** the percentage relative variation between BPR-MF and APR-MF metric values. Table 8.3 shows the accuracy and beyond-accuracy metrics presented in Section 8.3.1, while Table 8.4 shows the popularity bias ones. For each dataset, we report the model's recommendation performance with the best Re values in the set of hyper-parameters combinations presented in Section 8.3.1.

Analysis of Accuracy and Beyond-Accuracy (RQ1)

Analyzing Table 8.3, we identify that APR tends to reduce the novelty and coverage values compared to the one measured on BPR. For the `ML100K` dataset, APR-MF improves Re, *Prec*, and nDCG by more than 2%, with a slight reduction of Nov, i.e., $\mathbf{R.V.}(Nov) = -0.27\%$. For `Last.fm`, we measured an $\mathbf{R.V.}(Rec) = +7.14\%$, while a $\mathbf{R.V.}(Cov_{\%}) = -9.5\%$. Similarly, the `ML1M` results improve the Re of 3.62% while decreasing 6% the recommendation novelty. The same behavior is even more noticeable for the `Yelp` results, where $\text{Cov}_{\%}$ got a reduction greater than the 42%. Hence, we argue that the APR could negatively influence the beyond-accuracy recommendation performance.

Table 8.3 Accuracy and beyond-accuracy metrics evaluated on top-50 recommendation lists. The ↑ means that a bigger metric value can be related to an amplification of popularity bias, ↓ means a reduction.

Dataset	Model	Accuracy			Beyond-Accuracy	
		Rec	*Prec*	*nDCG*	*Nov*	*Cov%*
ML100K	BPR-MF	0.3871	0.0077	0.1222	2.7653	71.22
	APR-MF	0.3966	0.0079	0.1260*	2.7577*	71.22*
	R.V.	**+2.47%**	**+2.47%**	**+3.15%**	**-0.27%**	**0.00%**
Last.fm	BPR-MF	0.0148	0.0003	0.0040	4.8170	20.02
	APR-MF	0.0159	0.0003	0.0042	4.7605	18.10
	R.V.	**+7.14%**	**+7.14%**	**3.92%**	**-1.17%**	**-9.59%**
Amazon	BPR-MF	0.2077	0.0042	0.0656	6.0431	99.37
	APR-MF	0.2130	0.0043	0.0687*	5.6805*	90.58*
	R.V.	**+2.58%**	**+2.58%**	**+4.63%**	**-6.00%**	**-8.85%**
ML1M	BPR-MF	0.2747	0.0055	0.0830	2.8576	76.19
	APR-MF	0.2846*	0.0057*	0.0868*	2.6794*	70.76*
	R.V.	**+3.62%**	**+3.62%**	**+4.68%**	**-6.24%**	**-7.13%**
Yelp	BPR-MF	0.0990	0.0020	0.0290	7.7969	77.71
	APR-MF	0.1065*	0.0021*	0.0311*	7.2165*	44.43*
	R.V.	**+7.55%**	**+7.55%**	**+7.49%**	**-7.44%**	**-42.83%**

* *indicates statistically significant results (p-value* ≤ 0.05*) using the paired-t-test.*

Analysis of Popularity Bias (RQ2)

Long-tail diversity. As expected by the analysis in Section 8.2.6, the three long-tail diversity scores get negative **R.V.** when comparing APR-MF with BPR-MF, its building block. Examining the ARP values, we identify that APR-MF results increase the recurrence of most popular items in the recommendation lists. For instance, the **R.V.**(ARP) = +23.18% on `Amazon`, +10.13% on `ML1M`, and +32.57% on `Yelp`. As stated by Abdollahpouri et al. [2], since the ARP is not a good measure of long-tail diversity when used only on its own, we also report APLT and ACLT. For both metrics, the **R.V.** values are negatives, a behavior consistent with the growth of ARP. This empirical evaluation further supports our argument that APR could amplify the popularity bias more than BPR.

Ranking-based statistical item under-recommendation. Table 8.4 also reports the RSP and REO. While Zhu et al. [247] used these metrics to study the bias on a different group of items based on categorical information, e.g., genres, we studied the

Table 8.4 Popularity bias metrics evaluated on top-50 recommendation lists. The ↑ means that a bigger metric value is related to an amplification of popularity bias, ↓ means a reduction.

Dataset	Model	Popularity Bias								
		Long-tail diversity			Ranking-based statistical item under-recommendation					
		ARP ↑	$APLT$ ↓	$ACLT$ ↓	P_{SH} ↑	P_{LT} ↓	RSP ↑	$\hat{P}_{SH}$ ↑	$\hat{P}_{LT}$ ↓	REO ↑
ML100K	BPR-MF	176.64	0.2890	14.4486	0.0953	0.0102	0.8058	0.5279	0.2167	0.4180
	APR-MF	177.33*	0.2841*	14.2068*	0.0959*	0.0101*	0.8099*	0.5549*	0.2048	0.4609
	R.V.	**+0.39%**	**-1.67%**	**-1.67%**	**+0.68%**	**-1.67%**	**+0.51%**	**+5.11%**	**-5.49%**	**+10.26%**
Last.fm	BPR-MF	110.77	0.0094	0.4704	0.0141	0.0000	0.9948	0.0985	0.0046	0.9116
	APR-MF	114.06*	0.0069*	0.3451*	0.0141*	0.0000*	0.9962*	0.1061	0.0046	0.9176
	R.V.	**+2.96%**	**-26.63%**	**-26.63%**	**+0.25%**	**-26.63%**	**+0.14%**	**7.69%**	**0.00%**	**+0.66%**
Amazon	BPR-MF	106.59	0.3541	17.7055	0.0625	0.0086	0.7572	0.3469	0.1045	0.5371
	APR-MF	131.30*	0.2829*	14.1471*	0.0694*	0.0069*	0.8191*	0.3595	0.1045	0.5496
	R.V.	**+23.18%**	**-20.10%**	**-20.10%**	**+11.02%**	**-20.10%**	**+8.17%**	**+3.63%**	**0.00%**	**+2.34%**
ML1M	BPR-MF	1,072.48	0.1819	9.0952	0.0512	0.0030	0.8907	0.3850	0.1108	0.5531
	APR-MF	1,181.12*	0.1405*	7.0262*	0.0538*	0.0023*	0.9184*	0.4089*	0.1001*	0.6067*
	R.V.	**+10.13%**	**-22.75%**	**-22.75%**	**+5.06%**	**-22.75%**	**+3.12%**	**+6.19%**	**-9.67%**	**+9.69%**
Yelp	BPR-MF	204.64	0.1428	7.1398	0.0083	0.0003	0.9198	0.1552	0.0215	0.7566
	APR-MF	271.30*	0.0585*	2.9264*	0.0091*	0.0001*	0.9693*	0.1752*	0.0115*	0.8773
	R.V.	**+32.57%**	**-59.01%**	**-59.01%**	**+9.83%**	**-59.01%**	**+5.38%**	**+12.93%**	**-46.72%**	**+15.95%**

** indicates statistically significant results (p-value ≤ 0.05) using the paired-t-test.*

items divided into the short-head and long-tail groups (see Section 8.2.6 for further details). Consistent with the previous findings, RSP and REO values grew up when employing APR. For instance, **R.V.**$(RSP) = +3.12\%$ and **R.V.**(REO) $= +9.69\%$ measured on the `ML1M` datasets, show that the recommendations are biased towards the short-head items. Variation even bigger on the experiments on the `Yelp` dataset, e.g., **R.V.**(REO) $= +15.95\%$. Finally, the comparison between $(P_{SH}, \hat{P}_{SH})$ and $(P_{LT}, \hat{P}_{LT})$ pairs of measures reveal that APR worsened the already discriminatory behavior of BPR on the popular items. For example, the P_{SH} value is 17 times higher than P_{LT} in BPR-MF, while it is 23 times higher in APR-MF for the results on the `ML1M` dataset. Similarly, the same ratio increases by more than three times in the `Yelp` dataset.

Before we explore the effects of APR hyper-parameters in Section 8.3.2, we try to connect the results observed in Tables 8.3 and 8.4 together with the dataset characteristics reported in Table 8.2. An interesting finding that we could observe is that `Yelp`, the dataset with the lowest density (0.0011), is the one on which APR had the highest impact of bias amplification and beyond-accuracy performance reduction. Simultaneously, `ML1M` and `ML100K` -the two denser datasets- show less evident performance worsening. Indeed, suppose we order the datasets from the smallest to the highest density, we have the following relation $Yelp < Last.fm < Amazon < ML1M < ML100K$. The same order is also present for the $\text{Cov}_{\%}$ values and the APLT **R.V.** - except for a position swap between `ML1M` and `Amazon`.

Study of α and ϵ (RQ3)

To further study the effect of APR, in this section, we investigate the variation of the adversarial perturbation budget (ϵ) and regularization coefficient (α). Figure 8.3 reports the Re and Nov values evaluated on the (α, ϵ)-model pairs that got the best Re. Additionally, we show the metric value of the most accurate BPR-MF model (straight black line).

The ranking accuracy achieved for the APR-MF models with $\epsilon = 1.0$ (straight red line) shows a behavior not comparable to the results obtained with $\epsilon < 1.0$. This behavior confirmed on the Nov plots reveals that the application of adversarial training with big magnitudes of the adversarial perturbation, e.g., $\epsilon \geq 1$ can considerably change the recommendation performance. The negative impact of $\epsilon \geq 1.0$ has also been observed by the original work that proposed the adversarial training strategy [115]. Extending the analysis to $\epsilon \in \{0.001, 0.01, 0.1\}$, we observe that the Re values follow the same pattern when we fix the dataset and vary α. For instance, APR-MF trained on `Amazon` leads to Re values higher than the BPR-MF in the combination $\alpha = 0.01$ and each $\epsilon < 1.0$. The APR accuracy improvements on BPR are verified for each (α, ϵ)-combination on the `ML100K` dataset (see Figure 8.3a). Then, we observe the value of APR in the case of the `Yelp` dataset in Figure 8.3e. In this case, we can see that APR with $\epsilon = 1.0$ could produce a model more accurate than the one learned by BPR. Since there is not a clear common pattern across the models trained on the five tested datasets, we conjecture that the performance of APR could depend on the structural and distributional characteristics of the dataset.

Similar to the findings on Re, the beyond-accuracy values measured with $\alpha = 0.01$ and $\epsilon < 1.0$ have patterns that vary with the dataset. From the novelty plots in Figure 8.3, we extract two findings. First, the (α, ϵ)-combinations where the APR accuracy performance is higher than the BPR have no correspondence on the cases where APR can lead to $Nov_{\%}$ improvements on BPR. Second, the APR models' novelty values are mostly lower than those measured on the BPR-MF model. These two observations are in line with the reduction of beyond-accuracy metrics shown in Table 8.3, and the strict connection between their reduction in the case of amplification of the popularity bias as argued by [46].

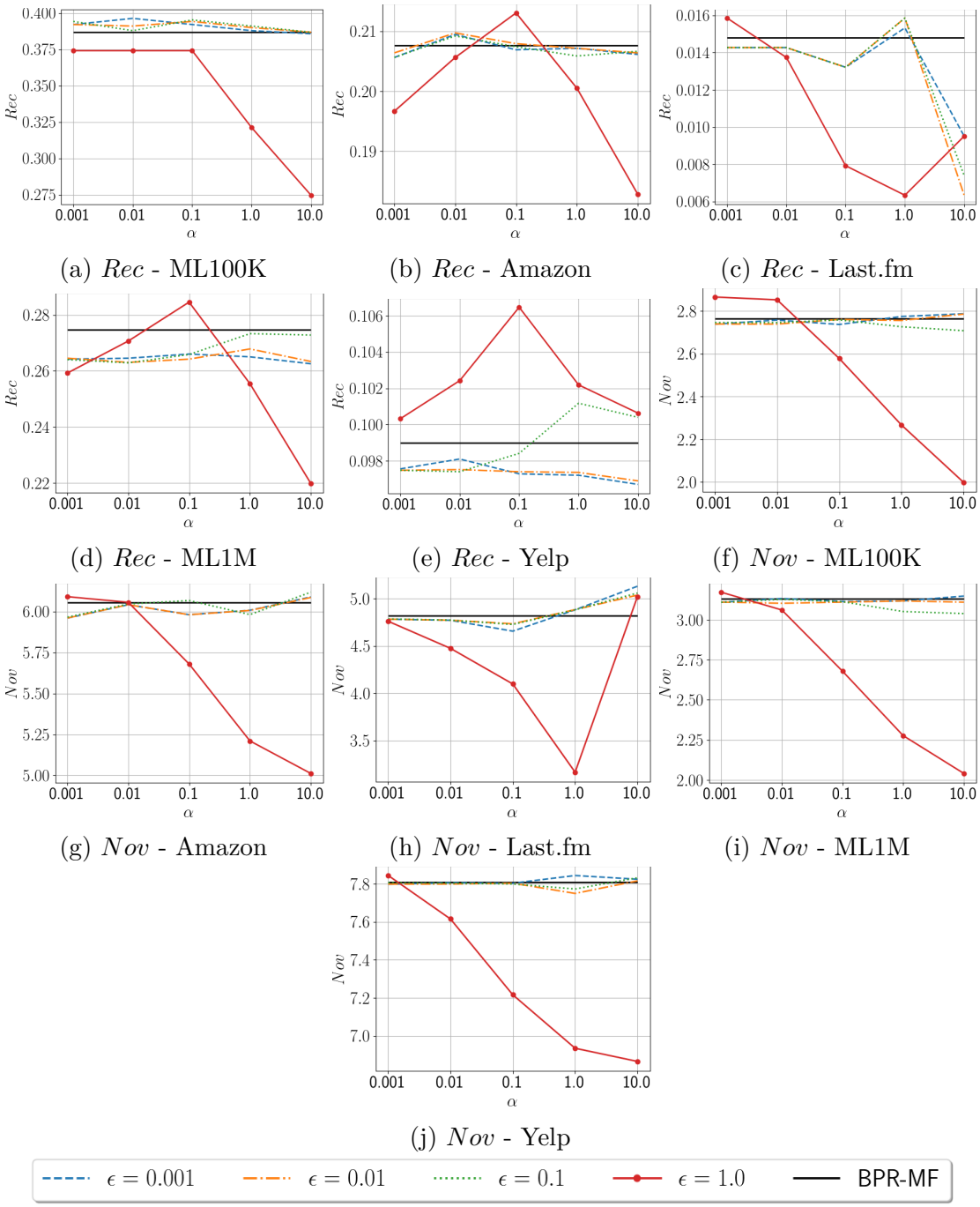


Fig. 8.3 Plots of the *Rec* (on the top) and *Rec* metrics on y-axis by varying α on x-axis.

8.4 Related Work

We now report on the literature related to applying adversarial learning techniques in the recommendation domain and the critical recommendation model features of beyond-accuracy performance and amplification of biases.

8.4.1 Models and Evaluation of AML in RSs

Adversarial Machine Learning (AML) is the field of study of the security of ML models. While the research on the injection of hand-engineered fake profiles has characterized the last twenty years of security investigation of RSs [48], the recent years view an increase of interest toward AML techniques [146, 115, 82]. The literature has been focused on three main classes of AML applications: (i) injection of adversarial perturbations on model parameters [115, 58, 209], (ii) adversarial attacks on the side information, e.g., items' images [85, 67, 154], and (iii) AML-optimized data poisoning attacks [146, 65] and defenses [92, 153]. The contribution of this chapter falls in the first class. Here, He et al. [115] proposed the pioneering application of AML for the item recommendation task. They reported the serious vulnerability of BPR-MF when the model parameters were adversarially perturbed. Additionally, the authors extended BPR with an adversarial training procedure, named adversarial personalized ranking (APR), as an effective defensive countermeasure. This work inspired a series of robustness studies on other core ML models and recommendation tasks. For instance, Tang et al. [209] applied the vulnerability study and proposed the APR defense to a visual-based RS for fashion recommendation. Yuan et al. [235, 234] investigated the robustification benefits of APR on a class of deep learning recommenders, the collaborative auto-encoder. Chen and Li [58] adopted the same approach to tensor-factorization models. Tran et al. [211] used APR for automatic playlist continuation. Manotumruksa and Yilmaz [156] implemented APR on a self-attention sequential recommender. In the literature of AML-RS for this class of attacks and defenses, the robustification analysis has been performed for the recommendation accuracy, leaving the beyond-accuracy evaluations as a completely low-investigated research field studied in this chapter.

8.4.2 Beyond-Accuracy and Popularity Bias in RSs

Due to the large impact of RSs in the society [35, 36], a huge research effort has been dedicated to beyond-accuracy objectives [44]. For instance, studying whether the suggested items are novel and cover the complete catalog, and proposing methods to mitigate several types of biases [60], e.g., selection bias[193], exposure bias [164], and popularity bias [1, 3]. Indeed, biases could lower the recommendation quality [51, 46]. In particular, the *popularity bias* is responsible for the "rich-get-richer" Matthew effect on RSs. In fact, a popularity-biased RS tends to recommend the most popular items, named *short-head items*, more than the less interacted ones, called *long-tail* [167].

Controlling and mitigating popularity biases has attracted massive interest in recent years. For instance, Abdollahpouri et al. [1] proposed both a regularization framework [1] and a re-ranking algorithm [2] to increase the coverage of long-tail items in the recommendation lists and reduce the bias amplification. Jannach et al. [130] proposed to reduce the popularity bias by sampling the training triplets of pairwise models including a user, an interacted (positive) item, and a not-interacted (negative) item, where the negative one is less popular than the unobserved item. Boratto et al. [46] integrated a balanced negative sampling technique with a novel objective function that reduces the biased correlation between the popularity of products and the user-item relevance score. In addition, BPR, the building block of the APR approach under our investigation, has been proved to amplify the recommendation lists' popularity-biased. For instance, Mansoury et al. [157] empirically identified that BPR is affected by a potent bias propagation phenomenon, Zhu et al. [247] measured the vulnerability of BPR to item under-recommendation bias, Boratto et al. [46] studied and connected the BPR item popularity bias to the low beyond-accuracy measures, e.g., novelty and coverage. The importance of beyond-accuracy evaluations [133], and the related amplification of popularity bias, motivated our extensive investigation on the APR optimization framework.

8.5 Summary

The current chapter has formally investigated the user of adversarial personalized ranking (APR) to robustify model-based recommendation algorithms. This technique is extensively used in many new recommendation models due to possible improvements in the robustness and accuracy of the models. While there has been much focus in investigating its efficacy in getting accuracy improvements in several recommendation tasks and domains, the assessment of APR effects on the beyond-accuracy evaluations has been under-investigated despite their importance on the recommendation quality and effectiveness. This chapter has proposed theoretical modeling of the APR learning characteristics starting from its building-block formulation, the Bayesian personalized ranking (BPR) optimization framework. We have formally identified that APR could be affected by a phenomenon of popularity bias amplification within a consistent reduction of beyond-accuracy performance. Then, we have identified that APR amplifies the popularity bias following a learning pattern that we named *wine-glass phenomenon*. The phenomenon confirmed that APR performs more positive gradient updates on short-head items than long-tail ones, with a difference in magnitude more conspicuous

than the one measured on BPR. Additionally, we have experimentally compared APR and BPR performance on MF recommenders trained on five standard recommendation datasets. We have also confirmed the theoretical findings of the APR amplification bias by measuring both beyond-accuracy and popularity bias performance worsening by varying the (α, ϵ) pairs of hyper-parameters. Considering the importance of APR as the first and popular technique to robustify the model parameters of model-based recommendation models, we consider it necessary to investigate novel robustification strategies and improve the existing one limiting the APR demonstrated tendency in worsening recommendation quality to present accurate, but also diverse, novel and more minor popularity-biased recommendations.

Chapter 9

Conclusions

The existence of adversarial examples has limited the areas in which deep learning can be applied in many tasks like computer vision, natural language processing, and speech recognition. Recently, adversarial samples have been demonstrated to effectively destroy the integrity and availability of recommendation models. In this dissertation, we have investigated three main areas of adversarial studies: (i) hand-engineered injection of fake profiles, (ii) adversarial perturbation of content data in multimedia settings, and (iii) minimal-sized perturbation on model parameters which are inside the three main areas of research of adversarial learning in recommendation task as shown in our published literature review [82] and book chapter [20].

Regarding the first research area of study, i.e., **hand-engineered injection of fake profiles**, we have contributed to the proposal of a regression-based framework to interpret the dataset characteristics that can influence the robustness of collaborative recommenders to the hand-engineered poisoning of the user-item recorded interactions. We have demonstrated in Chapter 3 that this tool can significantly support system designers to understand how to mitigate adversarial effects by stimulating the activeness of customers in interacting with the platform. Then, in Chapter 4, we have presented a set of novel **attack strategies that employ public available semantic information** like knowledge graphs to build powerful fake profiles that can have a dramatic impact on the reliability of the recommender system. The evidence of these limits has opened novel challenges in proposing novel defenses under these novel malevolent settings.

Then, we have focused our research contribution to investigating integrity and availability issues on **adversarial perturbation of content data in multimedia settings**. Chapter 5 has been devoted to presenting a part of our contribution in demonstrating that adversaries can easily break a visual recommender by uploading an adversarial sample of products (poisoning settings). We have verified that state-of-the-

art adversarial robustification strategies in both recommendation and computer vision domains are almost unuseful to protect the quality of recommendation lists. We have produced a set of empirical observations from which further studies can be based. For instance, we have shown that what makes an adversarial sample more impactful on the recommendation performance is the variations of the feature values used in the learning phase (e.g., the higher difference between original and adversarial samples makes the attack very strong independently of the attack success in misclassifying the classifiers used as feature extractor). Later, Chapter 6 has been dedicated to the presentation of our novel **defense proposal to protect visual recommenders against test time adversarial attacks**. In particular, we have tested whether a denoiser autoencoder is trained to preserve the original image characteristics and recommendation performance.

Finally, in the last part of the dissertation, we have investigated two main aspects of the **minimal-sized perturbation on model parameters**. First, we have explored novel adversarial strategies to build adversarial noises that we have demonstrated to completely break the recommendation quality of model-based recommenders. Chapter 7 have been dedicated to proposing our multi-step adversarial perturbation strategies that have opened novel perspectives on the robustness evaluation of model recommenders. That is: *the existence of slight variations of model parameters that make completely unuseful a recommender model in producing personalized recommendations is applicable in reality? If so, how can we make the model more robust?* A first defense solution that we verified to protect from iterative perturbations partially is the adversarial training strategy (see APR in Section 2.3.2). Interestingly, in Chapter 8, we have shown that subsequent articles have started to use APR as another optimization framework to improve the recommendation accuracy. Here, we have verified, via a **formal analysis of adversarial training for recommender systems**, that the motivation of this accuracy improvement could be related to a phenomenon of amplification of popularity bias that also motivates a drastic reduction on beyond-accuracy metric values (e.g., novelty, coverage, and diversity). This creates an avenue for future work exploring how to resist strong adversarial perturbations by preserving accuracy and beyond-accuracy performance.

Taken together, the research contributions presented in this dissertation pave the way towards more robust recommender systems. Novel (and applicable) attack strategies will be the basis of recommenders protected against adversaries. The limits of the existing defenses can motivate further research to guarantee the most reliable recommendations. The attention towards a complete analysis of the recommendation quality of defended models should motivate defense proposals that also consider beyond-

accuracy aspects. We hope that the content of this dissertation will serve as a stepping stone to build robust recommendation systems against adversaries.

References

[1] Abdollahpouri, H., Burke, R., and Mobasher, B. (2017). Controlling popularity bias in learning-to-rank recommendation. In *RecSys*, pages 42–46. ACM.

[2] Abdollahpouri, H., Burke, R., and Mobasher, B. (2019). Managing popularity bias in recommender systems with personalized re-ranking. In *FLAIRS Conference*, pages 413–418. AAAI Press.

[3] Abdollahpouri, H., Mansoury, M., Burke, R., and Mobasher, B. (2020). The connection between popularity bias, calibration, and fairness in recommendation. In *RecSys*, pages 726–731. ACM.

[4] Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.*, 17(6):734–749.

[5] Adomavicius, G. and Zhang, J. (2012). Impact of data characteristics on recommender systems performance. *ACM Trans. Management Inf. Syst.*, 3(1):3:1–3:17.

[6] Aggarwal, C. C. (2016a). Attack-resistant recommender systems. In *Recommender Systems*, pages 385–410. Springer.

[7] Aggarwal, C. C. (2016b). *Recommender Systems - The Textbook*. Springer.

[8] Aktukmak, M., Yilmaz, Y., and Uysal, I. (2019). Quick and accurate attack detection in recommender systems through user attributes. In *RecSys*, pages 348–352. ACM.

[9] Alonso, S., Bobadilla, J., Ortega, F., and Moya, R. (2019). Robust model-based reliability approach to tackle shilling attacks in collaborative filtering recommender systems. *IEEE Access*, 7:41782–41798.

[10] Anelli, V. W., Bellini, V., Di Noia, T., Bruna, W. L., Tomeo, P., and Di Sciascio, E. (2017a). An analysis on time- and session-aware diversification in recommender systems. In *UMAP*, pages 270–274. ACM.

[11] Anelli, V. W., Bellini, V., Di Noia, T., and Di Sciascio, E. (2020a). Knowledge-aware interpretable recommender systems. In *Knowledge Graphs for eXplainable Artificial Intelligence*, volume 47 of *Studies on the Semantic Web*, pages 101–124. IOS Press.

[12] Anelli, V. W., Bellogín, A., Deldjoo, Y., Di Noia, T., and Merra, F. A. (2021a). MSAP: multi-step adversarial perturbations on recommender systems embeddings. In Bell, E. and Keshtkar, F., editors, *Proceedings of the Thirty-Fourth International Florida Artificial Intelligence Research Society Conference, North Miami Beach, Florida, USA, May 17-19, 2021*.

[13] Anelli, V. W., Bellogín, A., Ferrara, A., Malitesta, D., Merra, F. A., Pomo, C., Donini, F. M., and Di Noia, T. (2021b). Elliot: A comprehensive and rigorous framework for reproducible recommender systems evaluation. In *SIGIR*, pages 2405–2414. ACM.

[14] Anelli, V. W., Bellogín, A., Ferrara, A., Malitesta, D., Merra, F. A., Pomo, C., Donini, F. M., and Di Noia, T. (2021c). How to perform reproducible experiments in the elliot recommendation framework: data processing, model selection, and performance evaluation. In *IIR*, CEUR Workshop Proceedings. CEUR-WS.org.

[15] Anelli, V. W., Bellogín, A., Ferrara, A., Malitesta, D., Merra, F. A., Pomo, C., Donini, F. M., and Di Noia, T. (2021d). V-elliot: Design, evaluate and tune visual recommender systems. In *RecSys*. ACM.

[16] Anelli, V. W., Deldjoo, Y., Di Noia, T., Di Sciascio, E., and Merra, F. A. (2020b). Sasha: Semantic-aware shilling attacks on recommender systems exploiting knowledge graphs. In *ESWC*, volume 12123 of *Lecture Notes in Computer Science*, pages 307–323. Springer.

[17] Anelli, V. W., Deldjoo, Y., Di Noia, T., Malitesta, D., and Merra, F. A. (2021e). A study of defensive methods to protect visual recommendation against adversarial manipulation of images. In *SIGIR*. ACM.

[18] Anelli, V. W., Deldjoo, Y., Di Noia, T., Malitesta, D., and Merra, F. A. (2021f). A study of defensive methods to protect visual recommendation against adversarial manipulation of images. In *SIGIR*, pages 1094–1103. ACM.

[19] Anelli, V. W., Deldjoo, Y., Di Noia, T., and Merra, F. A. (2020c). Adversarial learning for recommendation: Applications for security and generative tasks - concept to code. In *RecSys*, pages 738–741. ACM.

[20] Anelli, V. W., Deldjoo, Y., Di Noia, T., and Merra, F. A. (2021g). Adversarial recommender systems: Attack, defense, and advances. In *Third Edition of Recommender Systems Handbook*. Springer.

[21] Anelli, V. W., Deldjoo, Y., Di Noia, T., Merra, F. A., Acciani, G., and Di Sciascio, E. (2020d). Knowledge-enhanced shilling attacks for recommendation. In *SEBD*, volume 2646 of *CEUR Workshop Proceedings*, pages 310–317. CEUR-WS.org.

[22] Anelli, V. W. and Di Noia, T. (2019). 2nd workshop on knowledge-aware and conversational recommender systems - kars. In *CIKM*, pages 3001–3002. ACM.

[23] Anelli, V. W., Di Noia, T., Di Sciascio, E., Malitesta, D., and Merra, F. A. (2021h). Adversarial attacks against visual recommendation: an investigation on the influence of items' popularity. In *OHARS@RecSys*, CEUR Workshop Proceedings. CEUR-WS.org.

[24] Anelli, V. W., Di Noia, T., Di Sciascio, E., Ragone, A., and Trotta, J. (2019). How to make latent factors interpretable by feeding factorization machines with knowledge graphs. In *ISWC (1)*, volume 11778 of *Lecture Notes in Computer Science*, pages 38–56. Springer.

[25] Anelli, V. W., Di Noia, T., Di Sciascio, E., Ragone, A., and Trotta, J. (2020). Semantic interpretation of top-n recommendations. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1.

[26] Anelli, V. W., Di Noia, T., Lops, P., and Di Sciascio, E. (2017b). Feature factorization for top-n recommendation: From item rating to features relevance. In *RecSysKTL*, volume 1887 of *CEUR Workshop Proceedings*, pages 16–21. CEUR-WS.org.

[27] Anelli, V. W., Di Noia, T., Malitesta, D., and Merra, F. A. (2020a). Assessing perceptual and recommendation mutation of adversarially-poisoned visual recommenders (short paper). In *DP@AI*IA*, volume 2776 of *CEUR Workshop Proceedings*, pages 49–56. CEUR-WS.org.

[28] Anelli, V. W., Di Noia, T., and Merra, F. A. (2021i). A formal analysis of recommendation quality of adversarially-trained recommenders. In *CIKM 2021:30th ACM International Conference on Information and Knowledge Management · 1-5 November 2021, Online*. ACM.

[29] Anelli, V. W., Di Noia, T., and Merra, F. A. (2021j). The idiosyncratic effects of adversarial training on bias in personalized recommendation learning. In *RecSys 2021: Fifteenth ACM Conference on Recommender Systems (RecSys '21), September 27-October 1, 2021, Amsterdam, Netherlands*. ACM.

[30] Anelli, V. W., Leone, R. D., Di Noia, T., Lukasiewicz, T., and Rosati, J. (2020b). Combining RDF and SPARQL with cp-theories to reason about preferences in a linked data setting. *Semantic Web*, 11(3):391–419.

[31] Angioni, S., Salatino, A. A., Osborne, F., Recupero, D. R., and Motta, E. (2020). Integrating knowledge graphs for analysing academia and industry dynamics. In *ADBIS/TPDL/EDA Workshops*, volume 1260 of *Communications in Computer and Information Science*, pages 219–225. Springer.

[32] Apté, C., Liu, B., Pednault, E. P. D., and Smyth, P. (2002). Business applications of data mining. *Commun. ACM*, 45(8):49–53.

[33] Athalye, A., Carlini, N., and Wagner, D. A. (2018). Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*.

[34] Backstrom, L. and Leskovec, J. (2011). Supervised random walks: predicting and recommending links in social networks. In *WSDM*, pages 635–644. ACM.

[35] Baeza-Yates, R. (2018). Bias on the web. *Commun. ACM*, 61(6):54–61.

[36] Baeza-Yates, R. (2020). Bias in search and recommender systems. In *RecSys*, page 2. ACM.

[37] Beigi, G., Mosallanezhad, A., Guo, R., Alvari, H., Nou, A., and Liu, H. (2020). Privacy-aware recommendation with private-attribute protection using adversarial learning. In *WSDM*.

[38] Belleau, F., Nolin, M., Tourigny, N., Rigault, P., and Morissette, J. (2008). Bio2rdf: Towards a mashup to build bioinformatics knowledge systems. *J. Biomed. Informatics*, 41(5):706–716.

[39] Bellogín, A., Castells, P., and Cantador, I. (2017). Statistical biases in information retrieval metrics for recommender systems. *Inf. Retr. J.*, 20(6):606–634.

[40] Bennett, J., Lanning, S., et al. (2007). The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. Citeseer.

[41] Bhatia, S., Dwivedi, P., and Kaur, A. (2018). That's interesting, tell me more! finding descriptive support passages for knowledge graph relationships. In *International Semantic Web Conference (1)*, volume 11136 of *Lecture Notes in Computer Science*, pages 250–267. Springer.

[42] Bhaumik, R., Williams, C., Mobasher, B., and Burke, R. (2006). Securing collaborative filtering against malicious attacks through anomaly detection. In *ITWP 2006*.

[43] Biggio, B., Corona, I., Fumera, G., Giacinto, G., and Roli, F. (2011). Bagging classifiers for fighting poisoning attacks in adversarial classification tasks. In *MCS*, volume 6713 of *Lecture Notes in Computer Science*, pages 350–359. Springer.

[44] Bobadilla, J., Ortega, F., Hernando, A., and Gutiérrez, A. (2013). Recommender systems survey. *Knowl. Based Syst.*, 46:109–132.

[45] Bollacker, K. D., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD Conference*, pages 1247–1250. ACM.

[46] Boratto, L., Fenu, G., and Marras, M. (2021). Connecting user and item perspectives in popularity debiasing for collaborative recommendation. *Inf. Process. Manag.*, 58(1):102387.

[47] Bordes, A., Usunier, N., García-Durán, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *NIPS*, pages 2787–2795.

[48] Burke, R., O'Mahony, M. P., and Hurley, N. J. (2015). Robust collaborative recommendation. In *Recommender Systems Handbook*, pages 961–995. Springer.

[49] Burke, R. D. (2007). Hybrid web recommender systems. In *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 377–408. Springer.

[50] Cai, Y. and Zhu, D. (2019). Trustworthy and profit: A new value-based neighbor selection method in recommender systems under shilling attacks. *Decision Support Systems*, 124:113112.

[51] Cañamares, R. and Castells, P. (2018). Should I follow the crowd?: A probabilistic analysis of the effectiveness of popularity in recommender systems. In *SIGIR*, pages 415–424. ACM.

[52] Candillier, L., Meyer, F., and Boullé, M. (2007). Comparing state-of-the-art collaborative filtering systems. In *MLDM*, volume 4571 of *Lecture Notes in Computer Science*, pages 548–562. Springer.

[53] Cantador, I., Brusilovsky, P., and Kuflik, T. (2011). 2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011). In *Proceedings of the 5th ACM conference on Recommender systems*, RecSys 2011, New York, NY, USA. ACM.

[54] Cao, J., Wu, Z., Mao, B., and Zhang, Y. (2013). Shilling attack detection utilizing semi-supervised learning method for collaborative recommender system. *World Wide Web*, 16(5-6):729–748.

[55] Cao, Y., Chen, X., Yao, L., Wang, X., and Zhang, W. E. (2020). Adversarial attacks and detection on reinforcement learning-based interactive recommender systems. In *SIGIR*, pages 1669–1672. ACM.

[56] Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I. J., Madry, A., and Kurakin, A. (2019). On evaluating adversarial robustness. *CoRR*, abs/1902.06705.

[57] Carlini, N. and Wagner, D. A. (2017). Towards evaluating the robustness of neural networks. In *SP 2017*.

[58] Chen, H. and Li, J. (2019a). Adversarial tensor factorization for context-aware recommendation. In *RecSys*, pages 363–367. ACM.

[59] Chen, H. and Li, J. (2019b). Data poisoning attacks on cross-domain recommendation. In *CIKM*, pages 2177–2180. ACM.

[60] Chen, J., Dong, H., Wang, X., Feng, F., Wang, M., and He, X. (2020). Bias and debias in recommender system: A survey and future directions. *CoRR*, abs/2010.03240.

[61] Chen, J., Zhang, H., He, X., Nie, L., Liu, W., and Chua, T. (2017). Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention. In *SIGIR*. ACM.

[62] Chen, L., Xu, Y., Xie, F., Huang, M., and Zheng, Z. (2019). Data poisoning attacks on neighborhood-based recommender systems. *CoRR*, abs/1912.04109.

[63] Chirita, P., Nejdl, W., and Zamfir, C. (2005). Preventing shilling attacks in online recommender systems. In *WIDM*, pages 67–74. ACM.

[64] Chong, X., Li, Q., Leung, H., Men, Q., and Chao, X. (2020). Hierarchical visual-aware minimax ranking based on co-purchase data for personalized recommendation. In *WWW 2020*.

[65] Christakopoulou, K. and Banerjee, A. (2019). Adversarial attacks on an oblivious recommender. In *RecSys*, pages 322–330. ACM.

[66] Cochez, M., Declerck, T., de Melo, G., Anke, L. E., Fetahu, B., Gromann, D., Kejriwal, M., Koutraki, M., Lécué, F., Palumbo, E., and Sack, H., editors (2018). *Proceedings of the First Workshop on Deep Learning for Knowledge Graphs and Semantic Technologies (DL4KGS) co-located with the 15th Extended Semantic Web Conerence (ESWC 2018), Heraklion, Crete, Greece, June 4, 2018*, volume 2106 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[67] Cohen, R., Shalom, O. S., Jannach, D., and Amir, A. (2021). A black-box attack model for visually-aware recommender systems. In *WSDM*, pages 94–102. ACM.

[68] Covington, P., Adams, J., and Sargin, E. (2016). Deep neural networks for youtube recommendations. In *RecSys*, pages 191–198. ACM.

[69] Cremonesi, P., Koren, Y., and Turrin, R. (2010). Performance of recommender algorithms on top-n recommendation tasks. In *RecSys*, pages 39–46. ACM.

[70] Cremonesi, P., Tripodi, A., and Turrin, R. (2011). Cross-domain recommender systems. In *ICDM Workshops*, pages 496–503. IEEE Computer Society.

[71] Dai, Q., Shen, X., Zhang, L., Li, Q., and Wang, D. (2019). Adversarial training methods for network embedding. In *WWW*, pages 329–339. ACM.

[72] Das, A., Datar, M., Garg, A., and Rajaram, S. (2007). Google news personalization: scalable online collaborative filtering. In *WWW*, pages 271–280. ACM.

[73] De Candia, G., Di Noia, T., Di Sciascio, E., and Merra, F. A. (2021). Amflp: Adversarial matrix factorization-based link predictor in social graphs. In *SEBD 2021: The 29th Italian Symposium on Advanced Database Systems, September 5-9, 2021, Pizzo Calabro (VV), Italy*. CEUR Workshop Proceedings.

[74] Deldjoo, Y., Anelli, V. W., Zamani, H., Kouki, A. B., and Di Noia, T. (2019a). Recommender systems fairness evaluation via generalized cross entropy. In *RMSE@RecSys*.

[75] Deldjoo, Y., Dacrema, M. F., Constantin, M. G., Eghbal-zadeh, H., Cereda, S., Schedl, M., Ionescu, B., and Cremonesi, P. (2019b). Movie genome: alleviating new item cold start in movie recommendation. *User Model. User Adapt. Interact.*, 29(2):291–343.

[76] Deldjoo, Y., Di Noia, T., Di Sciascio, E., and Merra, F. A. (2020a). How dataset characteristics affect the robustness of collaborative recommendation models. In *SIGIR*, pages 951–960. ACM.

[77] Deldjoo, Y., Di Noia, T., Di Sciascio, E., and Merra, F. A. (2021a). A regression framework to interpret the robustness of recommender systems against shilling attacks. In *IIR*, CEUR Workshop Proceedings. CEUR-WS.org.

[78] Deldjoo, Y., Di Noia, T., Malitesta, D., and Merra, F. A. (2021b). A study on the relative importance of convolutional neural networks in visually-aware recommender systems. In *CVPR Workshops*, pages 3961–3967. Computer Vision Foundation / IEEE.

[79] Deldjoo, Y., Di Noia, T., Malitesta, D., and Merra, F. A. (2022). Leveraging content-style item representation for visual recommendation. In *The 44th European Conference on Information Retrieval*.

[80] Deldjoo, Y., Di Noia, T., and Merra, F. A. (2019c). Assessing the impact of a user-item collaborative attack on class of users. In *ImpactRS@RecSys*, volume 2462 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[81] Deldjoo, Y., Di Noia, T., and Merra, F. A. (2020b). Adversarial machine learning in recommender systems (aml-recsys). In *WSDM*, pages 869–872. ACM.

[82] Deldjoo, Y., Di Noia, T., and Merra, F. A. (2021c). A survey on adversarial recommender systems: From attack/defense strategies to generative adversarial networks. *ACM Comput. Surv.*, 54(2):35:1–35:38.

[83] Deldjoo, Y., Schedl, M., Cremonesi, P., and Pasi, G. (2020c). Recommender systems leveraging multimedia content. *ACM Comput. Surv.*, 53(5):106:1–106:38.

[84] Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Li, F. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society.

[85] Di Noia, T., Malitesta, D., and Merra, F. A. (2020). Taamr: Targeted adversarial attack against multimedia recommender systems. In *DSN Workshops*, pages 1–8. IEEE.

[86] Di Noia, T., Mirizzi, R., Ostuni, V. C., Romito, D., and Zanker, M. (2012). Linked open data to support content-based recommender systems. In *Proc. of the 8th Int. Conf. on Semantic Systems*, pages 1–8. ACM.

[87] Di Noia, T., Ostuni, V. C., Tomeo, P., and Di Sciascio, E. (2016). Sprank: Semantic path-based ranking for top-N recommendations using linked open data. *ACM TIST*, 8(1):9:1–9:34.

[88] Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., and Zhang, W. (2014). Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *KDD*, pages 601–610. ACM.

[89] Du, Y., Fang, M., Yi, J., Xu, C., Cheng, J., and Tao, D. (2019). Enhancing the robustness of neural collaborative filtering systems under malicious attacks. *IEEE Trans. Multimedia*, 21(3):555–565.

[90] Ekstrand, M. D., Riedl, J., and Konstan, J. A. (2011). Collaborative filtering recommender systems. *Found. Trends Hum. Comput. Interact.*, 4(2):175–243.

[91] Elsweiler, D., Trattner, C., and Harvey, M. (2017). Exploiting food choice biases for healthier recipe recommendation. In *SIGIR*, pages 575–584. ACM.

[92] Entezari, N., Al-Sayouri, S. A., Darvishzadeh, A., and Papalexakis, E. E. (2020). All you need is low (rank): Defending against adversarial attacks on graphs. In *WSDM*, pages 169–177. ACM.

[93] Fang, M., Gong, N. Z., and Liu, J. (2020). Influence function based data poisoning attacks to top-n recommender systems. In *WWW*, pages 3019–3025. ACM / IW3C2.

[94] Fang, M., Yang, G., Gong, N. Z., and Liu, J. (2018). Poisoning attacks to graph-based recommender systems. In *ACSAC*, pages 381–392. ACM.

[95] Feng, F., He, X., Tang, J., and Chua, T. (2019). Graph adversarial training: Dynamically regularizing based on graph structure. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1.

[96] Fernández-Tobías, I., Cantador, I., Tomeo, P., Anelli, V. W., and Di Noia, T. (2019). Addressing the user cold start with cross-domain collaborative filtering: exploiting item metadata in matrix factorization. *User Model. User Adapt. Interact.*, 29(2):443–486.

[97] Geng, X., Zhang, H., Bian, J., and Chua, T. (2015). Learning image and user features for recommendation in social networks. In *ICCV*, pages 4274–4282. IEEE Computer Society.

[98] Goldberg, D., Nichols, D. A., Oki, B. M., and Terry, D. B. (1992). Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70.

[99] Gomez-Uribe, C. A. and Hunt, N. (2016). The netflix recommender system: Algorithms, business value, and innovation. *ACM Trans. Management Inf. Syst.*, 6(4):13:1–13:19.

[100] Goodfellow, I. J., McDaniel, P. D., and Papernot, N. (2018). Making machine learning robust against adversarial inputs. *Commun. ACM*, 61(7):56–66.

[101] Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *ICLR (Poster)*.

[102] Grauman, K. (2020). Computer vision for fashion: From individual recommendations to world-wide trends. In *WSDM 2020*.

[103] Gu, S. and Rigazio, L. (2015). Towards deep neural network architectures robust to adversarial examples. In *ICLR (Workshop)*.

[104] Gunawardana, A. and Shani, G. (2015). Evaluating recommender systems. In *Recommender Systems Handbook*, pages 265–308. Springer.

[105] Gunes, I., Kaleli, C., Bilge, A., and Polat, H. (2014). Shilling attacks against recommender systems: a comprehensive survey. *Artif. Intell. Rev.*, 42(4):767–799.

[106] Guo, C., Rana, M., Cissé, M., and van der Maaten, L. (2018). Countering adversarial images using input transformations. In *ICLR 2018*.

[107] Guo, Q., Zhuang, F., Qin, C., Zhu, H., Xie, X., Xiong, H., and He, Q. (5555). A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge & Data Engineering*, (01).

[108] Hansen, C., Mehrotra, R., Hansen, C., Brost, B., Maystre, L., and Lalmas, M. (2021). Shifting consumption towards diverse content on music streaming platforms. In *WSDM*, pages 238–246. ACM.

[109] Harper, F. M. and Konstan, J. A. (2016). The movielens datasets: History and context. *TiiS*, 5(4):19:1–19:19.

[110] Hartig, O. (2017). Foundations of rdf⋆ and sparql⋆ (an alternative approach to statement-level metadata in RDF). In *AMW*, volume 1912 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[111] He, G., Li, J., Zhao, W. X., Liu, P., and Wen, J. (2020). Mining implicit entity preference from user-item interaction data for knowledge graph completion via adversarial learning. In *WWW*, pages 740–751. ACM / IW3C2.

[112] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE Computer Society.

[113] He, R. and McAuley, J. J. (2016a). Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *WWW 2016*.

[114] He, R. and McAuley, J. J. (2016b). VBPR: visual bayesian personalized ranking from implicit feedback. In *AAAI*, pages 144–150. AAAI Press.

[115] He, X., He, Z., Du, X., and Chua, T. (2018). Adversarial personalized ranking for recommendation. In *SIGIR*, pages 355–364. ACM.

[116] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T. (2017). Neural collaborative filtering. In *WWW*, pages 173–182. ACM.

[117] Heitmann, B. and Hayes, C. (2010). Using linked data to build open, collaborative recommender systems. In *AAAI Spring Symposium: Linked Data Meets Artificial Intelligence*. AAAI.

[118] Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. (2017). An algorithmic framework for performing collaborative filtering. *SIGIR Forum*, 51(2):227–234.

[119] Herlocker, J. L., Konstan, J. A., and Riedl, J. (2000). Explaining collaborative filtering recommendations. In *CSCW*, pages 241–250. ACM.

[120] Hidano, S. and Kiyomoto, S. (2020). Recommender systems robust to data poisoning using trim learning. In *ICISSP*, pages 721–724. SCITEPRESS.

[121] Hidasi, B., Karatzoglou, A., Baltrunas, L., and Tikk, D. (2016). Session-based recommendations with recurrent neural networks. In *ICLR (Poster)*.

[122] Hiemstra, D., Moens, M., Mothe, J., Perego, R., Potthast, M., and Sebastiani, F., editors (2021). *Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021, Proceedings, Part II*, volume 12657 of *Lecture Notes in Computer Science*. Springer.

[123] Hsu, C., Chung, H., and Huang, H. (2004). Mining skewed and sparse transaction data for personalized shopping recommendation. *Machine Learning*, 57(1-2):35–59.

[124] Hu, R., Guo, Y., Pan, M., and Gong, Y. (2019). Targeted poisoning attacks on social recommender systems. In *GLOBECOM*, pages 1–6. IEEE.

[125] Hu, Y., Yi, X., and Davis, L. S. (2015). Collaborative fashion recommendation: A functional tensor factorization approach. In *ACM Multimedia*, pages 129–138. ACM.

[126] Huang, Z., Chen, H., and Zeng, D. D. (2004). Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Trans. Inf. Syst.*, 22(1):116–142.

[127] Hug, N. (2017). Surprise, a Python library for recommender systems.

[128] Hulpus, I., Prangnawarat, N., and Hayes, C. (2015). Path-based semantic relatedness on linked data and its use to word and entity disambiguation. In *International Semantic Web Conference (1)*, volume 9366 of *Lecture Notes in Computer Science*, pages 442–457. Springer.

[129] Huo, Y., Wong, D. F., Ni, L. M., Chao, L. S., and Zhang, J. (2020). Knowledge modeling via contextualized representations for lstm-based personalized exercise recommendation. *Inf. Sci.*, 523:266–278.

[130] Jannach, D., Lerche, L., Kamehkhosh, I., and Jugovac, M. (2015). What recommenders recommend: an analysis of recommendation biases and possible countermeasures. *User Model. User Adapt. Interact.*, 25(5):427–491.

[131] Jannach, D., Zanker, M., Felfernig, A., and Friedrich, G. (2010). *Recommender Systems - An Introduction*. Cambridge University Press.

[132] Kallumadi, S. and Hsu, W. H. (2018). Interactive recommendations by combining user-item preferences with linked open data. In Mitrovic, T., Zhang, J., Chen, L., and Chin, D., editors, *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization, UMAP 2018, Singapore, July 08-11, 2018*, pages 121–125. ACM.

[133] Kaminskas, M. and Bridge, D. (2017). Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Trans. Interact. Intell. Syst.*, 7(1):2:1–2:42.

[134] Kang, W., Fang, C., Wang, Z., and McAuley, J. J. (2017). Visually-aware fashion recommendation and design with generative image models. In *ICDM*, pages 207–216. IEEE Computer Society.

[135] Katz, L. (1953). A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43.

[136] Kordan, S. B. and Kotov, A. (2018). Deep neural architecture for multi-modal retrieval based on joint embedding space for text and images. In *WSDM 2018*.

[137] Koren, Y. (2010). Factor in the neighbors: Scalable and accurate collaborative filtering. *TKDD*, 4(1):1:1–1:24.

[138] Koren, Y. and Bell, R. M. (2015). Advances in collaborative filtering. In *Recommender Systems Handbook*, pages 77–118.

[139] Koren, Y., Bell, R. M., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

[140] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *NeurIPS 2012*.

[141] Kula, M. (2015). Metadata embeddings for user and item cold-start recommendations. In *CBRecSys@RecSys 2015*.

[142] Kurakin, A., Goodfellow, I. J., and Bengio, S. (2017). Adversarial examples in the physical world. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net.

[143] Lam, S. K. and Riedl, J. (2004). Shilling recommender systems for fun and profit. In *WWW*, pages 393–402. ACM.

[144] Lee, K. and Lee, K. (2015). Escaping your comfort zone: A graph-based recommender system for finding novel recommendations among relevant items. *Expert Syst. Appl.*, 42(10):4851–4858.

[145] Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., and Bizer, C. (2015). Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

[146] Li, B., Wang, Y., Singh, A., and Vorobeychik, Y. (2016). Data poisoning attacks on factorization-based collaborative filtering. In *NIPS*, pages 1885–1893.

[147] Li, J., Xu, Z., Tang, Y., Zhao, B., and Tian, H. (2020a). Deep hybrid knowledge graph embedding for top-n recommendation. In *WISA*, volume 12432 of *Lecture Notes in Computer Science*, pages 59–70. Springer.

[148] Li, R., Wu, X., and Wang, W. (2020b). Adversarial learning to compare: Self-attentive prospective customer recommendation in location based social networks. In *WSDM*, pages 349–357. ACM.

[149] Liao, F., Liang, M., Dong, Y., Pang, T., Hu, X., and Zhu, J. (2018). Defense against adversarial attacks using high-level representation guided denoiser. In *CVPR*, pages 1778–1787. Computer Vision Foundation / IEEE Computer Society.

[150] Linden, G., Smith, B., and York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Comput.*, 7(1):76–80.

[151] Liu, D., Bai, T., Lian, J., Zhao, X., Sun, G., Wen, J., and Xie, X. (2019). News graph: An enhanced knowledge graph for news recommendation. In *KaRS@CIKM*, volume 2601 of *CEUR Workshop Proceedings*, pages 1–7. CEUR-WS.org.

[152] Liu, D., Sun, Y., Zhao, X., Zhang, G., and Liu, R. (2020). Adversarial training for session-based item recommendations. In *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, volume 9, pages 1162–1168.

[153] Liu, Y., Xia, X., Chen, L., He, X., Yang, C., and Zheng, Z. (2020). Certifiable robustness to discrete adversarial perturbations for factorization machines. In *SIGIR*, pages 419–428. ACM.

[154] Liu, Z. and Larson, M. (2021). Adversarial item promotion: Vulnerabilities at the core of top-n recommenders that use images to address cold start. In *Proceedings of the Web Conference 2021*, page 3590–3602, New York, NY, USA. Association for Computing Machinery.

[155] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *ICLR 2018*.

[156] Manotumruksa, J. and Yilmaz, E. (2020). Sequential-based adversarial optimisation for personalised top-n item recommendation. In *SIGIR*, pages 2045–2048. ACM.

[157] Mansoury, M., Abdollahpouri, H., Pechenizkiy, M., Mobasher, B., and Burke, R. (2020). Feedback loop and bias amplification in recommender systems. In *CIKM*, pages 2145–2148. ACM.

[158] Markwood, P. S. (2010). The long tail: Why the future of business is selling less of more. *Learn. Publ.*, 23(3):268–269.

[159] McAuley, J. J., Targett, C., Shi, Q., and van den Hengel, A. (2015). Image-based recommendations on styles and substitutes. In *SIGIR 2015*.

[160] McDaniel, P. D., Papernot, N., and Celik, Z. B. (2016). Machine learning in adversarial settings. *IEEE Secur. Priv.*, 14(3):68–72.

[161] Mobasher, B., Burke, R., Bhaumik, R., and Williams, C. (2005). Effective attack models for shilling item-based collaborative filtering systems. In *Proceedings of the WebKDD Workshop*, pages 13–23. Citeseer.

[162] Mobasher, B., Burke, R. D., Bhaumik, R., and Sandvig, J. J. (2007a). Attacks and remedies in collaborative recommendation. *IEEE Intelligent Systems*, 22(3):56–63.

[163] Mobasher, B., Burke, R. D., Bhaumik, R., and Williams, C. (2007b). Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. *ACM Trans. Internet Techn.*, 7(4):23.

[164] Morik, M., Singh, A., Hong, J., and Joachims, T. (2020). Controlling fairness and bias in dynamic learning-to-rank. In *SIGIR*, pages 429–438. ACM.

[165] Natarajan, S., Vairavasundaram, S., Natarajan, S., and Gandomi, A. H. (2020). Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data. *Expert Syst. Appl.*, 149:113248.

[166] Nayyeri, M., Vahdati, S., Zhou, X., Yazdi, H. S., and Lehmann, J. (2020). Embedding-based recommendations on scholarly knowledge graphs. In *ESWC*, volume 12123 of *Lecture Notes in Computer Science*, pages 255–270. Springer.

[167] Nikolov, D., Lalmas, M., Flammini, A., and Menczer, F. (2019). Quantifying biases in online information exposure. *J. Assoc. Inf. Sci. Technol.*, 70(3):218–229.

[168] Ning, X., Desrosiers, C., and Karypis, G. (2015). A comprehensive survey of neighborhood-based recommendation methods. In *Recommender Systems Handbook*, pages 37–76. Springer.

[169] Ning, X. and Karypis, G. (2012). Sparse linear methods with side information for top-n recommendations. In *Sixth ACM Conference on Recommender Systems, RecSys '12, Dublin, Ireland, September 9-13, 2012*, pages 155–162.

[170] Niu, W., Caverlee, J., and Lu, H. (2018). Neural personalized ranking for image recommendation. In *WSDM 2018*.

[171] Nunes, B. P., Dietze, S., Casanova, M. A., Kawase, R., Fetahu, B., and Nejdl, W. (2013). Combining a co-occurrence-based and a semantic measure for entity linking. In *ESWC*, volume 7882 of *Lecture Notes in Computer Science*, pages 548–562. Springer.

[172] Ojino, R. (2019). User's profile ontology-based semantic model for personalized hotel room recommendation in the web of things: student research abstract. In Hung, C. and Papadopoulos, G. A., editors, *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC 2019, Limassol, Cyprus, April 8-12, 2019*, pages 2314–2316. ACM.

[173] O'Mahony, M. P., Hurley, N. J., Kushmerick, N., and Silvestre, G. C. M. (2004). Collaborative recommendation: A robustness analysis. *ACM Trans. Internet Techn.*, 4(4):344–377.

[174] O'Mahony, M. P., Hurley, N. J., and Silvestre, G. C. M. (2005). Recommender systems: Attack types and strategies. In *AAAI*, pages 334–339. AAAI Press / The MIT Press.

[175] O'Mahony, M. P., Hurley, N. J., and Silvestre, G. C. (2003). An evaluation of the performance of collaborative filtering. In *14th Irish Artificial Intelligence and Cognitive Science (AICS 2003) Conference*. Citeseer.

[176] Palumbo, E., Monti, D., Rizzo, G., Troncy, R., and Baralis, E. (2020). entity2rec: Property-specific knowledge graph embeddings for item recommendation. *Expert Syst. Appl.*, 151:113235.

[177] Papernot, N., Faghri, F., Carlini, N., Goodfellow, I., Feinman, R., Xie, A. K. C., Sharma, Y., Brown, T., Roy, A., Matyasko, A., Behzadan, V., Hambardzumyan, K., Zhang, Z., Yi-LJuang, Li, Z., Sheatsley, R., Garg, A., Uesato, J., Gierke, W., Dong, Y., Berthelot, D., Hendricks, P., Rauber, J., and Long, R. (2018). Technical report on the cleverhans v2.1.0 adversarial examples library. *Corr 2018*.

[178] Papernot, N., McDaniel, P. D., Wu, X., Jha, S., and Swami, A. (2016). Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE Symposium on Security and Privacy*, pages 582–597. IEEE Computer Society.

[179] Park, D. H. and Chang, Y. (2019). Adversarial sampling and training for semi-supervised information retrieval. In *WWW*, pages 1443–1453. ACM.

[180] Paulheim, H. and Fürnkranz, J. (2012). Unsupervised generation of data mining features from linked open data. In *WIMS*, pages 31:1–31:12. ACM.

[181] Pazzani, M. J. and Billsus, D. (2007). Content-based recommendation systems. In *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 325–341. Springer.

[182] Prinz, K., Flexer, A., and Widmer, G. (2021). On end-to-end white-box adversarial attacks in music information retrieval. *Trans. Int. Soc. Music. Inf. Retr.*, 4(1):93.

[183] Quadrana, M., Cremonesi, P., and Jannach, D. (2018). Sequence-aware recommender systems. *ACM Comput. Surv.*, 51(4):66:1–66:36.

[184] Raghunathan, A., Xie, S. M., Yang, F., Duchi, J. C., and Liang, P. (2020). Understanding and mitigating the tradeoff between robustness and accuracy. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 7909–7919. PMLR.

[185] Rao, C. R. (1973). *Linear Statistical Inference and its Applications, Second Editon*. Wiley Series in Probability and Statistics. Wiley.

[186] Rendle, S. (2010). Factorization machines. In *ICDM 2010*.

[187] Rendle, S. and Freudenthaler, C. (2014). Improving pairwise learning for item recommendation from implicit feedback. In *WSDM*, pages 273–282. ACM.

[188] Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009). BPR: bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461. AUAI Press.

[189] Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L. (2010). Factorizing personalized markov chains for next-basket recommendation. In *WWW*. ACM.

[190] Ricci, F., Rokach, L., and Shapira, B. (2011). Introduction to recommender systems handbook. In *Recommender Systems Handbook*, pages 1–35.

[191] Ricci, F., Rokach, L., and Shapira, B., editors (2015). *Recommender Systems Handbook*. Springer.

[192] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *MICCAI (3)*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer.

[193] Saito, Y. (2020). Asymmetric tri-training for debiasing missing-not-at-random explicit feedback. In *SIGIR*, pages 309–318. ACM.

[194] Sang, L., Xu, M., Qian, S., and Wu, X. (2021). Knowledge graph enhanced neural collaborative recommendation. *Expert Syst. Appl.*, 164:113992.

[195] Schedl, M. (2016). The lfm-1b dataset for music retrieval and recommendation. In *ICMR*, pages 103–110. ACM.

[196] Shadbolt, N., O'Hara, K., Berners-Lee, T., Gibbins, N., Glaser, H., Hall, W., and m. c. schraefel (2012). Linked open government data: Lessons from data.gov.uk. *IEEE Intelligent Systems*, 27(3):16–24.

[197] Shafahi, A., Najibi, M., AmGhiasi, Xu, Z., Dickerson, J. P., Studer, C., Davis, L. S., GavTaylor, and Goldstein, T. (2019). Adversarial training for free! In *NeurIPS 2019*.

[198] Shi, D., Wang, T., Xing, H., and Xu, H. (2020). A learning path recommendation model based on a multidimensional knowledge graph framework for e-learning. *Knowl. Based Syst.*, 195:105618.

[199] Shi, Y., Larson, M. A., and Hanjalic, A. (2010). List-wise learning to rank with matrix factorization for collaborative filtering. In *RecSys*, pages 269–272. ACM.

[200] Shi, Y., Larson, M. A., and Hanjalic, A. (2014). 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.

[201] Shmueli, G. et al. (2010). To explain or to predict? *Statistical science*.

[202] Si, M. and Li, Q. (2020). Shilling attacks against collaborative recommender systems: a review. *Artif. Intell. Rev.*, 53(1):291–319.

[203] Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *ICLR*.

[204] Smith, B. and Linden, G. (2017). Two decades of recommender systems at amazon.com. *IEEE Internet Computing*, 21(3):12–18.

[205] Song, J., Li, Z., Hu, Z., Wu, Y., Li, Z., Li, J., and Gao, J. (2020). Poisonrec: An adaptive data poisoning framework for attacking black-box recommender systems. In *ICDE*, pages 157–168. IEEE.

[206] Steck, H. (2011). Item popularity and recommendation accuracy. In *RecSys*, pages 125–132. ACM.

[207] Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). Yago: a core of semantic knowledge. In *WWW*, pages 697–706. ACM.

[208] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. (2014). Intriguing properties of neural networks. In *ICLR*.

[209] Tang, J., Du, X., He, X., Yuan, F., Tian, Q., and Chua, T. (2020). Adversarial training towards robust multimedia recommender system. *IEEE Trans. Knowl. Data Eng.*, 32(5):855–867.

[210] Tanon, T. P., Weikum, G., and Suchanek, F. M. (2020). YAGO 4: A reason-able knowledge base. In *ESWC*, volume 12123 of *Lecture Notes in Computer Science*, pages 583–596. Springer.

[211] Tran, T., Sweeney, R., and Lee, K. (2019). Adversarial mahalanobis distance-based attentive song recommender for automatic playlist continuation. In *SIGIR*, pages 245–254. ACM.

[212] Tu, Z., Zhang, J., and Tao, D. (2019). Theoretical analysis of adversarial learning: A minimax approach. In *NeurIPS*.

[213] Vargas, S. (2014). Novelty and diversity enhancement and evaluation in recommender systems and information retrieval. In *SIGIR*, page 1281. ACM.

[214] Vargas, S. and Castells, P. (2011). Rank and relevance in novelty and diversity metrics for recommender systems. In *RecSys*.

[215] Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P. (2008). Extracting and composing robust features with denoising autoencoders. In *ICML*, volume 307 of *ACM International Conference Proceeding Series*, pages 1096–1103. ACM.

[216] Vorobeychik, Y. and Kantarcioglu, M. (2018). *Adversarial Machine Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

[217] Vrandecic, D. (2012). Wikidata: a new platform for collaborative data collection. In *WWW (Companion Volume)*, pages 1063–1064. ACM.

[218] Wang, H., Wang, N., and Yeung, D. (2015). Collaborative deep learning for recommender systems. In *KDD*, pages 1235–1244. ACM.

[219] Wang, H., Zhang, F., Wang, J., Zhao, M., Li, W., Xie, X., and Guo, M. (2019a). Exploring high-order user preference on the knowledge graph for recommender systems. *ACM Trans. Inf. Syst.*, 37(3):32:1–32:26.

[220] Wang, H., Zhao, M., Xie, X., Li, W., and Guo, M. (2019b). Knowledge graph convolutional networks for recommender systems. In *WWW*, pages 3307–3313. ACM.

[221] Wang, J., Fu, Z., Niu, M., Zhang, P., and Zhang, Q. (2020a). Multi-feedback pairwise ranking via adversarial training for recommender. *Chinese Journal of Electronics*, 29(4):615–622.

[222] Wang, J. and Han, P. (2020). Adversarial training-based mean bayesian personalized ranking for recommender system. *IEEE Access*, 8:7958–7968.

[223] Wang, S., Wang, Y., Tang, J., Shu, K., Ranganath, S., and Liu, H. (2017). What your images reveal: Exploiting visual contents for point-of-interest recommendation. In *WWW*, pages 391–400. ACM.

[224] Wang, T., Shi, D., Wang, Z., Xu, S., and Xu, H. (2020b). Mrp2rec: Exploring multiple-step relation path semantics for knowledge graph-based recommendations. *IEEE Access*, 8:134817–134825.

[225] Wang, X., He, X., Wang, M., Feng, F., and Chua, T. (2019c). Neural graph collaborative filtering. In *SIGIR*, pages 165–174. ACM.

[226] Weibo, H., Chuan, C., Yaomin, C., Zibin, Z., and Yunfei, D. (2021). Robust graph convolutional networks with directional graph adversarial training. *Applied Intelligence*.

[227] Wiyatno, R. R., Xu, A., Dia, O., and de Berker, A. (2019). Adversarial examples in modern machine learning: A review. *CoRR*, abs/1911.05268.

[228] Wu, Z., Liu, Y., Zhang, Q., Wu, K., Zhang, M., and Ma, S. (2019). The influence of image search intents on user behavior and satisfaction. In *WSDM 2019*.

[229] Xu, Y., Chen, L., Xie, F., Hu, W., Zhu, J., Chen, C., and Zheng, Z. (2020). Directional adversarial training for recommender systems. In *ECAI*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 553–560. IOS Press.

[230] Yang, G., Gong, N. Z., and Cai, Y. (2017). Fake co-visitation injection attacks to recommender systems. In *NDSS*.

[231] Yang, Z. and Dong, S. (2020). Hagerec: Hierarchical attention graph convolutional network incorporating knowledge graph for explainable recommendation. *Knowl. Based Syst.*, 204:106194.

[232] Yin, R., Li, K., Lu, J., and Zhang, G. (2019). Enhancing fashion recommendation with visual compatibility relationship. In *WWW 2019*.

[233] Yu, W., Zhang, H., He, X., Chen, X., Xiong, L., and Qin, Z. (2018). Aesthetic-based clothing recommendation. In Champin, P., Gandon, F. L., Lalmas, M., and Ipeirotis, P. G., editors, *WWW 2018*.

[234] Yuan, F., Yao, L., and Benatallah, B. (2019a). Adversarial collaborative auto-encoder for top-n recommendation. In *IJCNN*, pages 1–8. IEEE.

[235] Yuan, F., Yao, L., and Benatallah, B. (2019b). Adversarial collaborative neural network for robust recommendation. In *SIGIR*, pages 1065–1068. ACM.

[236] Yuan, F., Yao, L., and Benatallah, B. (2020). Exploring missing interactions: A convolutional generative adversarial network for collaborative filtering. In *CIKM*, pages 1773–1782. ACM.

[237] Yuan, X., He, P., Zhu, Q., and Li, X. (2019c). Adversarial examples: Attacks and defenses for deep learning. *IEEE Trans. Neural Networks Learn. Syst.*, 30(9):2805–2824.

[238] YZhang and Caverlee, J. (2019). Instagrammers, fashionistas, and me: Recurrent fashion recommendation with implicit visual influence. In *CIKM 2019*.

[239] Zhang, H., Li, Y., Ding, B., and Gao, J. (2020). Practical data poisoning attack against next-item recommendation. In *WWW*, pages 2458–2464. ACM / IW3C2.

[240] Zhang, Q., Hao, P., Lu, J., and Zhang, G. (2019a). Cross-domain recommendation with semantic correlation in tagging systems. In *IJCNN*, pages 1–8. IEEE.

[241] Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*.

[242] Zhang, S., Yao, L., Sun, A., and Tay, Y. (2019b). Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv.*, 52(1):5:1–5:38.

[243] Zhou, T., Kuscsik, Z., Liu, J.-G., Medo, M., Wakeling, J. R., and Zhang, Y.-C. (2010). Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences*, 107(10):4511–4515.

[244] Zhou, W., Wen, J., Qu, Q., Zeng, J., and Cheng, T. (2018). Shilling attack detection for recommender systems based on credibility of group users and rating time series. *PloS one*, 13(5):e0196533.

[245] Zhou, W., Wen, J., Xiong, Q., Gao, M., and Zeng, J. (2016). Svm-tia a shilling attack detection method based on svm and target item analysis in recommender systems. *Neurocomputing*, 210:197–205.

[246] Zhu, Z., Hu, X., and Caverlee, J. (2018). Fairness-aware tensor-based recommendation. In *CIKM*.

[247] Zhu, Z., Wang, J., and Caverlee, J. (2020). Measuring and mitigating item under-recommendation bias in personalized ranking systems. In *SIGIR*, pages 449–458. ACM.