Assessing the Impact of a User-Item Collaborative Attack on Class of Users*

Yashar Deldjoo
Polytechnic University of Bari
Bari, Italy
yashar.deldjoo@poliba.it

Tommaso Di Noia
Polytechnic University of Bari
Bari, Italy
tommaso.dinoia@poliba.it

Felice Antonio Merra†
Polytechnic University of Bari
Bari, Italy
felice.merra@poliba.it

Abstract

Collaborative Filtering (CF) models lie at the core of most recommendation systems due to their state-of-the-art accuracy. They are commonly adopted in e-commerce and online services for their impact on sales volume and/or diversity, and their impact on companies' outcome. However, CF models are only as good as the interaction data they work with. As these models rely on outside sources of information, counterfeit data such as user ratings or reviews can be injected by attackers to manipulate the underlying data and alter the impact of resulting recommendations, thus implementing a so-called shilling attack. While previous works have focused on evaluating shilling attack strategies from a global perspective paying particular attention to the effect of the size of attacks and attacker's knowledge, in this work we explore the effectiveness of shilling attacks under novel aspects. First, we investigate the effect of attack strategies crafted on a *target user* in order to push the recommendation of a low-ranking *item* to a higher position, referred to as *user-item attack*. Second, we evaluate the effectiveness of attacks in altering the impact of different CF models by contemplating the *class* of the target user, from the perspective of the richness of her profile (i.e., *slightly-active* v.s. *highly-active* user). Finally, similar to previous work we contemplate the *size of attack* (i.e., the amount of fake profiles injected) in examining their success.

The results of experiments on two widely used datasets in business and movie domains, namely Yelp and MovieLens, suggest that highly-active and slightly-active users exhibit contrasting behaviors in datasets with different characteristics.

1 Introduction and Related Work

Collaborative filtering (CF) models are a crucial component in many real-world recommendation services due to their state-of-the-art accuracy. Considering their widespread popularity and adoption in the industry, the output of these models can impact many decision qualities in different application scenarios [3, 16, 28]. The open nature of CF models, which rely on user-specified judgments (e.g., ratings or reviews) to build user profiles and compute recommendation, can be used in the hand of adversaries to manipulate the underlying data and affect the impact of recommendation, a phenomenon commonly referred to as shilling attacks [11, 19]. The attacker may manipulate the recommender for positive motivations, like outcomes improvement, or malicious, like reducing the user's loyalty to a competitor.

Presented at the ImpactRS workshop held in conjunction with the 13th ACM Conference on Recommender Systems (RecSys), 2019, in Copenhagen, Denmark.

†Authors are listed in alphabetical order. Corresponding author: Felice Antonio Merra (felice.merra@poliba.it).

In this direction, first works [15, 19, 24] focused on different profile injection strategies by analyzing and classifying them on the required effort and amount of attacker's knowledge to craft successful attacks. These works have been followed by multiple studies on the evaluation of the robustness [4, 7, 22] of different CF models and detection strategies [8, 18, 29]. The robustness analysis in surveys [11, 21] shows that Item-*k*NN is more robust than User-*k*NN and model-based CF are generally more resistant to shilling attacks than conventional nearest neighbor-based algorithms.

One common characteristic of the previous literature on shilling attacks on CF-RS is their focus on assessing the global impact of shilling attacks on different CF models by examining the success of attacks from the perspective of attacker's knowledge and the size of attack (i.e. the number of shilling profiles) [11]. In the present work instead, we investigate the effectiveness of an attack on a target-item of a target-user, namely *user-item attack*, with a novel point of attention focused on influence of the attack on the *classes of attacked users* in particular *highly-active (HA) user* and *slightly-active (SA) user*.

The application scenario for class-based study of attacks on RS may span in different domains. As an example, a restaurant owner may wish to diminish the trust on a target user of a competitor by pushing a low-ranked product *for the specific user*. The same argument can be made for new users. An attacker may be interested in pushing or nuking, particular products with the objective of modifying the impact of a recommender system in order to affect future interactions of the new user.

The leading reserach questions of this work are then:

- **RQ1:** From a global perspective, what is the impact of user-item attack on *classes of users* such as slightly-active and highly-active users?
  - Could attacks be tailored to have a higher impact on a particular class of users?
  - Which factors play a role on the impact of such an attack?
- **RQ2:** From a local perspective, how do CF recommendation model work differently under user-item attacks by looking to user-classes?

The remainder of the paper is structured as follows. Section 2 presents the evaluation protocol and datasets description we used in our experimental evaluation. Section 3 reports on the results and their discussion. Section 4 concludes the paper and introduces future perspectives.

2 User-Item attack modeling and evaluation

In this section, we discuss our evaluation protocol for a user-item attack modeling and the corresponding evaluation setup.

2.1 Evaluation Protocol

In order to test the effects of a user-item attack on attacked user classes, an extensive set of experiments has been carried out with respect to three dimensions: (i) the attack strategy (type and quantity of injected profiles), (ii) core CF recommendation model and (iii) the user classes. The experimental evaluation has been executed on two well-known datasets, MovieLens-1M (ML-1M) and Yelp (described in Section 2.2).

2.1.1 Attack Strategies. We have implemented two attack strategies to craft shilling profiles (SP) in order to model different level of attacker's capability. Given a user profile $P(u) = \{r_{i_1}, \ldots, r_{i_n}\}$ (consisting of a set of items rated by user u), we consider the items in $P(u)$ in the form of: *selected items* (I_S), *filler items* (I_F), *target item* (I_T) previously identified in [6], with $|I_S| + |I_F| + |I_T| = |P(u)|$. The items in the set I_F are selected randomly in order to obstruct detection of an SP while the only element in I_T is the item that the attacker wants to push, or nuke. Here we focus on two strategies to build I_S, which lies at the core of a shilling profile generation. The number of items in a shilling profile is close to the mean value of the number of rating in the dataset. We execute two types of attacks:

- **User-and-Model aware attack** (*UMA*) assumes a partial knowledge of some victim preferences. The attacker creates a new profile, called *seed profile*, on the system with these preferences and uses the recommendation systems to receive recommendations. The recommendations are then used to fill I_S with high ratings. This type of attack is inspired by the *probe attack* [2, 6, 11]. In the probe attack, the seed profile is created by the adversary and the recommendations generated by the recommender system are used to learn related items and their ratings in order to built up shilling profiles very similar to existing users in the system. These items constitute the 50% of each shilling profile.
- **User-Neighbor aware attack** (*UNA*) assumes that the attacker knows some users similar to the victim. We employ this attack by evaluating the *k-nearest neighbor* users of each victim[1] and selecting the most rated items in the neighbor in order to fill I_S. This attack is a modified version of the *bandwagon or popular attack* [25]. While the bandwagon attack sets high ratings on the popular items of the system; the proposed attack sets high ratings on the popular items inside the victim's neighborhood in order in order to inject profiles capable to influence more the victim-s recommendations.

We executed experiments with different size of injected profiles, which are classified in *small-size* attacks by averaging results of attacks with 2, 10, 20, 50 shilling profiles and *large-size* attacks by averaging attacks with 200 and 500 injected profiles.

2.1.2 CF Models. In our evaluation, we compared the vulnerability/robustness of the following CF models:

User-kNN [5]: user-based k-nearest-neighbor (kNN) method. In our experiments, we set the number of neighbors k to 20 [19].

Item-kNN [27]: item-based kNN method. Also in this case, the number of neighbors k has been set equal to 20.

[1]experiment setting: $k = 50$, *similarity metric* = cosine similarity.

BPR-SLIM [23]: Sparse LInear Method (SLIM) is an item-item model that models the estimation of unknown user-item rating as a regression problem. It learns a sparse aggregation coefficient matrix from aggregated users' preferences. This model allows the system to capture correlations between items. BPR-SLIM uses the BPR optimization criterion.[2]

BPR-MF [26]: This method uses matrix factorization (MF) as its underlying core predictor and optimizes it with Bayesian Personalized Ranking (BPR) objective function.

These CF models stand for state-of-the-art models for the item recommendation task, each using a different prediction concept, allowing us to study the impact of different attack strategies from multiple viewpoints.

2.1.3 User Classes. Given that CF models only rely on user preference scores (i.e., ratings) to compute recommendation, we hypothesize that it is relevant to investigate the impact of different attack strategies with respect to the victim user's level of activity, i.e. the richness of her profile, calculated on the basis of the number of ratings available in her profile. To this aim, we define two classes of users:

- **Highly-active (HA) users** are defined as users who have a number of ratings greater than the second quartile of the number of ratings for each user in the dataset.
- **Slightly-active (SA) users** are defined as users who have a number of ratings lower than the second quartile.

2.1.4 Evaluation Metric. Several metrics have already been proposed to evaluate malicious attacks. For example, [24] proposes the prediction shift (PS) which estimates the success of an attack by measuring the prediction difference before and after the attack [30]. It has been identified that a strong PS does not necessarily implies an effective attack result [20]. From the perspective of the attacker, the ideal goal in a push attack is to increase the chance of a desired item being recommended after the attack than before. We use a *modified version* of *Hit-Ratio* [17] to measure the fraction of successful attacks on a set of different user-item pairs.

Definition 1. *Let u be the user under attack and i be the targeted item that the attacker wants to push/appear in the top-k recommendations of u. Let top_u^k be the top-k recommendations of u. Let $\phi(i, top_u^k)$ be the function to evaluate the effectiveness on an attack on (u, i). If i is pushed in the top-k then $\phi(i, top_u^k) = 1$ (successful attack), otherwise $\phi(i, top_u^k) = 0$ (unsuccessful attack). Let S be the set of (u, i) user-item pairs under attack. $HR@k$ is defined as the fraction of successful attacks on each $(u, i) \in S$.*

$$HR@k = \frac{\sum_{(u,i)\in S} \phi(i, top_u^k)}{|S|} \tag{1}$$

where $|S|$ is the number of (u, i) pairs over which $HR@k$ is measured.

2.2 Data Descriptions

We conducted experiments on two well-known datasets, MovieLens 1M [12] and Yelp [13, 14]. The datasets represent different item

[2]The computation of the CF comparative models has been done with the publicly available software library MyMediaLite http://www.mymedialite.net/. We used default parameters for both BPR-MF and BPR-SLIM.

recommendation scenarios for movie and business domains and have data densities which are approximately 40 times different from each other. Table 1 summarizes the statistics of the two datasets (after pre-processing).

Table 1: Characteristics of the dataset used in the offline experiment: $|\mathcal{U}|$ is the number of users, $|\mathcal{I}|$ the number of items, $|\mathcal{R}|$ the number of ratings

Dataset	$\|\mathcal{U}\|$	$\|\mathcal{I}\|$	$\|\mathcal{R}\|$	$\frac{\|\mathcal{R}\|}{\|\mathcal{I}\|\cdot\|\mathcal{U}\|} \times 100$
ML-1M	6040	3706	1000209	4.468%
Yelp	5135	5163	24809	0.093%

MovieLens-1M: We used a million-sized version of the dataset ML-1M, which contain 1M ratings of users for items (movies). We used the original ML-1M dataset for the experiments without any filtering.

Yelp: This dataset contains ratings of users on businesses. We used the pre-processed version of the dataset provided by [13, 14] with 731K ratings of 25K users for 25K businesses. Given the large size of users and items from which item-item or user-similarities have to be computed, similar to [1] we extracted a random sample of 5K users and 5K items in order to speed up the experiments. The resulted dataset contains 24.8K ratings with data density (0.110%), which is comparable with the one before filtering (0.093%).

3 Results and Discussion

In order to validate the empirical impact of the under study attack types *on different classes of users*, an extensive set of experiments has been carried out with respect to the dimensions introduced in Section 2.1. The final results are presented in Table 2 and discussed from the following viewpoints:

- A *global analysis* of the impact of attacks on user classes (cf. Section 3.1)
- A *fine-grained analysis* of the impact of attacks on user classes by looking into the CF models and attack types. (cf. Section 3.2)

We present each of these analysis viewpoints in the following subsections.

3.1 Global impact of attacks on user classes

The goal of this analysis is to answer the first research question related to the global assessment on the effectiveness of user-item attack with respect to the identified users classes. We use the term global here, since in this analysis we would like to free our attention from the impact of attacks on CF models, attack quality (type) and/or quantity as they have been largely addressed in previous works [11, 21, 22]. Instead, we examine the impact of attacks on the dimension of user classes by looking into the aggregate mean values computed across CF models on the two datasets we adopted in our experimental evaluation.

A general observation for the results in Table 2 is that larger-size attacks reach higher level of effectiveness on both classes of users (highly-active and slightly-active) in comparison with smaller-size attacks. For example, on the Yelp dataset, the average HR@10 for UNA attack on highly-active users (across CF models) is 0.256 for small-size attacks, while it is 0.800 for large-size attacks, a difference of approximately three times. The same pattern of results is obtained in other experimental cases. These results are in line with those presented in previous works [21, 22].

Our objective here is to study the impact of different attack strategies on user classes. For this purpose, we define the variable $r = \frac{HR_{HA}}{HR_{SA}}$ and refer to it as *user-class attack impact* —i.e., the impact of an attack on highly-active users in comparison with slightly-active users. Different values for r are interpreted as in the following:

- $r = 1$: the attack has an *equal impact* on highly-active and slightly-active users.
- $r > 1$: the attack has an *unequal impact* on highly-active w.r.t slightly-active users. The impact of attack on *highly-active users* is *relatively* higher in comparison with slightly-active users.[3]
- $r < 1$: the attack has *an unequal impact* on highly-active v.s. slightly-active users. The impact of attack on *slightly-active users* is *relatively* higher in comparison with highly-active users.

It is obvious that the larger r deviates from the center point 1, the larger is the attack success in differentiating highly-active with respect to slightly-active users in one of the above-mentioned directions ($r < 1$ or $r > 1$). Before starting a deeper analysis of the results we highlight that the most interesting values are in the left portion of Table 2 (small-size attacks), because when the size of attack is larger the attack reaches the maximum effectiveness, $HR = 1$, independently of user classes.

By looking at the results for each attack size in Table 2, we can see that the average user-class impact $\bar{r}$ has a value higher than 1 for the Yelp dataset ($\bar{r} > 1$), while a value lower than 1 for the ML-1M dataset ($\bar{r} < 1$). These results show that *both attack types have an unequal impact on slightly-active vs highly-active users* as $r \neq 1$. However, *the class of users they have a larger impact on remains largely different and contrasting in the two datasets.*

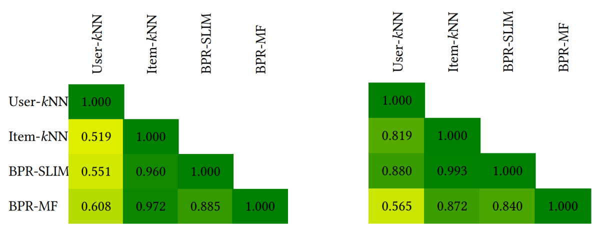
As an example, in Yelp and for UMA, one can note that for small-size attack $\bar{r} = 2.393$ and for large-size attack $\bar{r} = 1.832$, while the corresponding values on ML-1M are $\bar{r} = 0.658$ and $\bar{r} = 0.909$, respectively. This means that the impact of attacks on user classes is higher on highly-active users on the Yelp dataset ($\bar{r}$>1), differently from ML-1M ($\bar{r}$<1).

We conjecture that the above contrasting behaviors are directly linked with the characteristics of the datasets such as their sparsity. As shown in Table 1, Yelp dataset is approximately 40 times sparser than ML-1M and we consider this difference as the main/possible cause of the contrasting outcomes in tested datasets. We try to provide a possible explanation here. In the more sparse dataset (i.e., the Yelp dataset), users with a small number of ratings (*slightly-active users*) are more immune to attacks because they have a smaller support size of the user profile (i.e., the user profile is not rich enough for the attacker to be able to mimic it in a crafted way). In contrast, highly-active users are more immune to attack in ML-1M with higher density, because their recommendations rely on neighbors with (very) rich user profiles. Put it simply, the crafted

[3] This is equal to say, slightly-active users are relatively more immune to the attack w.r.t. highly-active users.

Table 2: HR@10 for *small-size* and *large-size* attacks with respect to the class of user, *slightly-active* and *highly-active*, and the CF model. The *user-class impact* r is the ratio of HR_{HA} value to HR_{SA}. (Abbreviations: HA → Highly Active, SA→Slightly Active)

Dataset	CF/Attack		Small-size attacks					Large-size attacks					
			U-*k*NN	I-*k*NN	BPR SLIM	BPR MF	**mean**	U-*k*NN	I-*k*NN	BPR SLIM	BPR MF	**mean**	**overall mean**
Yelp	**UMA**	SA	0.750	0.067	0.225	0.108	**0.288**	0.967	0.184	0.500	0.533	**0.546**	**0.417**
		HA	0.800	0.350	0.492	0.117	**0.440**	1.000	0.667	0.784	0.584	**0.758**	**0.599**
		r	*1.067*	*5.243*	*2.184*	*1.079*	**2.393**	*1.034*	*3.632*	*1.567*	*1.095*	**1.832**	**2.113**
	UNA	SA	0.850	0.625	0.792	0.400	**0.667**	1.000	0.834	1.000	1.000	**0.958**	**0.813**
		HA	0.875	0.742	0.850	0.433	**0.725**	1.000	0.850	1.000	1.000	**0.963**	**0.844**
		r	*1.029*	*1.186*	*1.074*	*1.082*	**1.093**	*1.000*	*1.020*	*1.000*	*1.000*	**1.005**	**1.049**
ML-1M	**UMA**	SA	0.302	0.155	0.267	0.121	**0.211**	0.897	0.086	0.586	0.328	**0.474**	**0.343**
		HA	0.092	0.108	0.159	0.125	**0.121**	0.383	0.150	0.350	0.284	**0.292**	**0.206**
		r	*0.303*	*0.698*	*0.593*	*1.037*	**0.658**	*0.427*	*1.744*	*0.597*	*0.866*	**0.909**	**0.783**
	UNA	SA	0.621	0.302	0.595	0.164	**0.420**	1.000	0.897	1.000	0.811	**0.927**	**0.673**
		HA	0.459	0.133	0.250	0.183	**0.256**	1.000	0.800	0.800	0.600	**0.800**	**0.528**
		r	*0.739*	*0.442*	*0.421*	*1.121*	**0.680**	*1.000*	*0.892*	*0.800*	*0.740*	**0.858**	**0.769**



(a) HR@10 Slightly-active Users (b) HR@10 Highly-active Users

Figure 1: Heat-map of Correlation Coefficient (ρ) of different measures between CF models for *small-size attacks*: (a) HR@10 on Slightly-active Users, (b) HR@10 on Highly-active Users.

attacks need to use a large number of profiles to be able to alter recommendation for the target user.

The insight on sparsity is an important indication that data characteristics are playing a role in the effectiveness of attacks and it motivates further research in this direction.

3.2 Fine-grained analysis of the impact of attacks on user classes

The goal of this analysis is to study how different CF models behave against the attacks: which ones have similar performance and which ones have a different performance. This study resembles previous work on shilling attacks on CF models. However, we take into account the impact of attack on user classes in this study as well.

Instead of individual CF models performances and attack types, we compute the pairwise Pearson correlation between each pair of analyzed CF models. Figure 1 indicates a strong correlation on HR@10 between BPR-SLIM and Item-*k*NN ($\rho = 0.960$ in Figure 1a and $\rho = 0.993$ in Figure 1b). We justify this value by the fact that both CF models exploit the item-item similarity computation. Looking at the correlation values for User-*k*NN in Figure 1, one can observe a slightly lower correlation in the case of *slightly-active-users* with respect to other models. We think that this phenomenon comes from the fact that tested attack are based on user preferences which gain good effect also with small-size attacks. For instance, HR@10 for Yelp on slightly-active users (0.750 and 0.850) is higher than the mean values with other models for both attack (mean = 0.288 and 0.440). We can also observe an interesting behavior when we compare ρ of BPR-MF with BPR-SLIM and Item-*k*NN. Figure 1 (a) and (b) show that HR@10 on both classes of attacked users is highly correlated ($\rho \geq 0.840$). Finally, results in Table 2 show that BPR-MF is the model that is less influenced by user-classes because the *user-impact factor* is close to 1 for each class of users and attacks.

4 Conclusion and Future Work

This work investigates the effect of user-item attacks on classes of users. Particularly, we investigated the effectiveness of attacks from a global and local perspective by varying the quality and quantity of attacks, the target user class and the collaborative filtering recommendation model.

Experimental results on Yelp and MovieLens datasets indicate that for Yelp dataset slightly-active users are more immune to shilling attacks than highly-active users, a characteristic that is in contrast with the results on MovieLens dataset where highly-active users are more immune than slightly-active users. As datasets have a very different sparsity (Yelp is approximately 40 time more sparse than MovieLens) we will move our future works in analyzing the effectiveness of dataset properties under different attack scenarios. From a local perspective, we evidence that BPR-MF is less influenced than other models when varying user-class and attack types. On the other hand, BPR-SLIM and Item-*k*NN have shown similar behavior related to the effect of attacks on user classes. In future, we also plan to extend our study by considering more datasets from different domains, exploring in an extensive way the influence of dataset properties, such as sparsity, user and item skewness, rating variance, on the effectiveness of different type of attacks. Also, it is of our interest to consider the impact of various shilling attack types on CF models using item content as side information [9, 10].

These studies give important insights on the impact of shilling attacks on recommender systems and provide clues on how to reduce their effectiveness by working on datasets characteristics.

References

[1] Gediminas Adomavicius and Jingjing Zhang. 2015. Improving Stability of Recommender Systems: A Meta-Algorithmic Approach. *IEEE Trans. Knowl. Data Eng.* 27, 6 (2015), 1573–1587. https://doi.org/10.1109/TKDE.2014.2384502

[2] Charu C. Aggarwal. 2016. *Recommender Systems - The Textbook.* Springer. https://doi.org/10.1007/978-3-319-29659-3

[3] Xavier Amatriain and Justin Basilico. 2015. Recommender Systems in Industry: A Netflix Case Study. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer, 385–419. https://doi.org/10.1007/978-1-4899-7637-6_11

[4] Runa Bhaumik, Chad Williams, Bamshad Mobasher, and Robin Burke. 2006. Securing collaborative filtering against malicious attacks through anomaly detection. In *Proceedings of the 4th Workshop on Intelligent Techniques for Web Personalization (ITWP'06), Boston*, Vol. 6. 10.

[5] John S. Breese, David Heckerman, and Carl Myers Kadie. 1998. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *UAI '98: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, University of Wisconsin Business School, Madison, Wisconsin, USA, July 24-26, 1998.* 43–52. https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=231&proceeding_id=14

[6] Robin Burke, Bamshad Mobasher, Roman Zabicki, and Runa Bhaumik. 2005. Identifying attack models for secure recommendation. *Beyond Personalization* 2005 (2005).

[7] Zunping Cheng and Neil Hurley. 2010. Robust Collaborative Recommendation by Least Trimmed Squares Matrix Factorization. In *22nd IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2010, Arras, France, 27-29 October 2010 - Volume 2.* 105–112. https://doi.org/10.1109/ICTAI.2010.90

[8] Paul-Alexandru Chirita, Wolfgang Nejdl, and Cristian Zamfir. 2005. Preventing shilling attacks in online recommender systems. In *Seventh ACM International Workshop on Web Information and Data Management (WIDM 2005), Bremen, Germany, November 4, 2005.* 67–74. https://doi.org/10.1145/1097047.1097061

[9] Yashar Deldjoo, Maurizio Ferrari Dacrema, Mihai Gabriel Constantin, Hamid Eghbal-zadeh, Stefano Cereda, Markus Schedl, Bogdan Ionescu, and Paolo Cremonesi. 2019. Movie genome: alleviating new item cold start in movie recommendation. *User Model. User-Adapt. Interact.* 29, 2 (2019), 291–343. https://doi.org/10.1007/s11257-019-09221-y

[10] Yashar Deldjoo, Markus Schedl, Paolo Cremonesi, and Gabriella Pasi. 2018. Content-Based Multimedia Recommendation Systems: Definition and Application Domains. In *Proceedings of the 9th Italian Information Retrieval Workshop, Rome, Italy, May, 28-30, 2018.* http://ceur-ws.org/Vol-2140/paper15.pdf

[11] Ihsan Gunes, Cihan Kaleli, Alper Bilge, and Huseyin Polat. 2014. Shilling attacks against recommender systems: a comprehensive survey. *Artif. Intell. Rev.* 42, 4 (2014), 767–799. https://doi.org/10.1007/s10462-012-9364-9

[12] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *TiiS* 5, 4 (2016), 19:1–19:19. https://doi.org/10.1145/2827872

[13] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial Personalized Ranking for Recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018.* 355–364. https://doi.org/10.1145/3209978.3209981

[14] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast Matrix Factorization for Online Recommendation with Implicit Feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016.* 549–558. https://doi.org/10.1145/2911451.2911489

[15] Shyong K. Lam and John Riedl. 2004. Shilling recommender systems for fun and profit. In *Proceedings of the 13th international conference on World Wide Web, WWW 2004, New York, NY, USA, May 17-20, 2004.* 393–402. https://doi.org/10.1145/988672.988726

[16] Greg Linden, Brent Smith, and Jeremy York. 2003. Industry Report: Amazon.com Recommendations: Item-to-Item Collaborative Filtering. *IEEE Distributed Systems Online* 4, 1 (2003).

[17] Bhaskar Mehta and Wolfgang Nejdl. 2008. Attack resistant collaborative filtering. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008.* 75–82. https://doi.org/10.1145/1390334.1390350

[18] Bhaskar Mehta and Wolfgang Nejdl. 2009. Unsupervised strategies for shilling detection and robust collaborative filtering. *User Model. User-Adapt. Interact.* 19, 1-2 (2009), 65–97. https://doi.org/10.1007/s11257-008-9050-4

[19] Bamshad Mobasher, Robin Burke, Runa Bhaumik, and Chad Williams. 2005. Effective attack models for shilling item-based collaborative filtering systems. Citeseer.

[20] Bamshad Mobasher, Robin Burke, and Jeff J Sandvig. 2006. Model-based collaborative filtering as a defense against profile injection attacks. In *AAAI*, Vol. 6. 1388.

[21] Bamshad Mobasher, Robin D. Burke, Runa Bhaumik, and Jeff J. Sandvig. 2007. Attacks and Remedies in Collaborative Recommendation. *IEEE Intelligent Systems* 22, 3 (2007), 56–63. https://doi.org/10.1109/MIS.2007.45

[22] Bamshad Mobasher, Robin D. Burke, Runa Bhaumik, and Chad Williams. 2007. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. *ACM Trans. Internet Techn.* 7, 4 (2007), 23. https://doi.org/10.1145/1278366.1278372

[23] Xia Ning and George Karypis. 2011. SLIM: Sparse Linear Methods for Top-N Recommender Systems. In *11th IEEE International Conference on Data Mining, ICDM 2011, Vancouver, BC, Canada, December 11-14, 2011.* 497–506. https://doi.org/10.1109/ICDM.2011.134

[24] Michael P. O'Mahony, Neil J. Hurley, Nicholas Kushmerick, and Guenole C. M. Silvestre. 2004. Collaborative recommendation: A robustness analysis. *ACM Trans. Internet Techn.* 4, 4 (2004), 344–377. https://doi.org/10.1145/1031114.1031116

[25] Michael P. O'Mahony, Neil J. Hurley, and Guenole C. M. Silvestre. 2005. Recommender Systems: Attack Types and Strategies. In *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA.* 334–339. http://www.aaai.org/Library/AAAI/2005/aaai05-053.php

[26] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009.* 452–461. https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1630&proceeding_id=25

[27] Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001.* 285–295. https://doi.org/10.1145/371920.372071

[28] Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR)* 47, 1 (2014), 3.

[29] Zhihai Yang and Zhongmin Cai. 2017. Detecting abnormal profiles in collaborative filtering recommender systems. *J. Intell. Inf. Syst.* 48, 3 (2017), 499–518. https://doi.org/10.1007/s10844-016-0424-5

[30] Mi Zhang, Jie Tang, Xuchen Zhang, and Xiangyang Xue. 2014. Addressing cold start in recommender systems: A semi-supervised co-training algorithm. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval.* ACM, 73–82.