Artif Intell Rev (2014) 42:767–799
DOI 10.1007/s10462-012-9364-9

# Shilling attacks against recommender systems: a comprehensive survey

**Ihsan Gunes · Cihan Kaleli · Alper Bilge · Huseyin Polat**

Published online: 2 November 2012


**Abstract** Online vendors employ collaborative filtering algorithms to provide recommendations to their customers so that they can increase their sales and profits. Although recommendation schemes are successful in e-commerce sites, they are vulnerable to shilling or profile injection attacks. On one hand, online shopping sites utilize collaborative filtering schemes to enhance their competitive edge over other companies. On the other hand, malicious users and/or competing vendors might decide to insert fake profiles into the user-item matrices in such a way so that they can affect the predicted ratings on behalf of their advantages. In the past decade, various studies have been conducted to scrutinize different shilling attacks strategies, profile injection attack types, shilling attack detection schemes, robust algorithms proposed to overcome such attacks, and evaluate them with respect to accuracy, cost/benefit, and overall performance. Due to their popularity and importance, we survey about shilling attacks in collaborative filtering algorithms. Giving an overall picture about various shilling attack types by introducing new classification attributes is imperative for further research. Explaining shilling attack detection schemes in detail and robust algorithms proposed so far might open a lead to develop new detection schemes and enhance such robust algorithms further, even propose new ones. Thus, we describe various attack types and introduce new dimensions for attack classification. Detailed description of the proposed detection and robust recommendation algorithms are given. Moreover, we briefly explain evaluation of the proposed schemes. We conclude the paper by discussing various open questions.

**Keywords** Shilling · Profile injection · Push/nuke attacks · Collaborative filtering · Robustness · Attack detection

I. Gunes · C. Kaleli · A. Bilge · H. Polat (✉)
Computer Engineering Department, Anadolu University, 26470 Eskisehir, Turkey
e-mail: polath@anadolu.edu.tr

## 1 Introduction

With increasing popularity of the Internet, online shopping has also been receiving increasing attention. Many customers prefer shopping (buy and/or sell various products) online over the Internet via various e-commerce sites. Due to the increasing number of users utilizing online vendors and number of items available online, it becomes a challenge to choose the right product without wasting too much time. To help their customers and surpass such challenge, many e-commerce sites use collaborative filtering (CF) algorithms. CF schemes help customers select the products that they might like given a set of possible items.

In a typical CF system, an $n \times m$ user-item matrix is created, where $n$ users' preferences about $m$ products are represented as ratings, either numeric or binary. To obtain a prediction for a target item $q$ or a sorted list of items that might be liked, an active user $a$ sends her known ratings and a query to the system. CF system estimates similarities between $a$ and each user in the database, forms a neighborhood by selecting the best similar users, and estimate a prediction ($p_{aq}$) or a recommendation list (top-$N$ recommendation) using a CF algorithm (Herlocker et al. 2004).

In order to not to frustrate their customers, online vendors utilizing CF systems must provide high quality recommendations efficiently. In other words, there are basically two major goals that traditional CF systems should achieve. Namely, they are called accuracy and online performance. Accuracy in this context means that the estimated predictions should be as close as possible to their true withheld values. Offering inaccurate recommendations might lead angry customers who might decide to shop over alternative e-commerce sites. Online vendors should also produce predictions to their customers during online interactions without wasting their time. Waiting constantly for a prediction might make customers frustrated. Inaccurate recommendations and poor online performance might cause e-commerce sites lose competitive edge over their competing companies. Therefore, it is critical for online vendors to provide accurate predictions efficiently.

CF systems can achieve the abovementioned goals if they collect high quality data. It is almost impossible to estimate accurate predictions from low quality data. Data collected for CF purposes might be vulnerable against some attacks. Malicious users and/or rival companies might try to insert fake profiles into user-item matrices in order to affect the predicted ratings and/or diminish the performance of the system on behalf of their goals. Some attacks might intend to increase the popularity of some targeted items (referred to as the *push attack*) while some others might aim to decrease the popularity of some targeted items (referred to as the *nuke attack*) (Mobasher et al. 2007b). Due to the inserted fake profiles, quality of the data diminishes while amount of available data increases. Low quality or noisy data make accuracy worse while augmented data due to inserted fake profiles make online performance worse.

For the overall success of CF schemes, it is imperative to handle shilling attacks. Due to its importance, researchers have been giving increasing attention to such attacks. Since it is almost impossible to prevent shilling attacks, in the literature, some researchers focus on shilling attack detection schemes. Some of them study shilling attacks and their types while the others scrutinize how to develop robust CF algorithms or enhance the robustness of CF algorithms against profile injection attacks. The researchers also evaluate various attacks using benchmark data sets and some perform cost-benefit analysis.

In the literature, shilling attacks are classified according to intend and amount of knowledge required to attack a system (Mobasher et al. 2007a). According to intend, they are grouped as push and nuke attacks while according to required knowledge, they are classified as low and high knowledge attacks. However, in addition to intend and knowledge attributes,

the attacks might be classified according to rating types, application, and CF algorithms. With increasing popularity of privacy, various privacy-preserving collaborative filtering (PPCF) schemes have been proposed (Polat and Du 2005). Although the researchers focus on shilling attacks planned against CF algorithms, such attacks might be performed against PPCF schemes, as well. Thus, they should open a new pavement to scrutinize PPCF algorithms with respect to shilling attacks. Moreover, although there are some studies surveying shilling attacks against CF algorithms, they come short discussing such attacks in the last five years. There is no comprehensive survey discussing shilling attacks, detection and robust algorithms, and their evaluation in detail.

In this study, we present a survey about shilling attacks against various CF algorithms. We investigate various studies with respect to shilling attacks. We describe attributes for classification shilling attacks and introduce new ones. In addition, major research directions (attack types, attack detection, robust algorithms, and cost/benefit analysis) are investigated. Since evaluation takes attention of most of the researchers, we give detail description of evaluation methodology, benchmark data sets, evaluation measures, and briefly discuss cost in terms of shilling attacks.

The paper is structured, as follows: In Sect. 2, we briefly discuss related studies, which focus on surveying about shilling attacks. After discussing various shilling attack types in detail in Sect. 3, we study major research areas about profile injection attacks in Sect. 4. We then discuss the related studies in terms of evaluation by explaining benchmark data sets and evaluation metrics in Sect. 5. In Sect. 6, we give a brief discussion about new directions. Finally, we conclude our paper and give some future directions in Sect. 7.

## 2 Related work

CF schemes are deployed commonly by e-commerce sites to entice customers and they are publicly available. However, due to the mechanism that they utilize to produce recommendations, they are not strictly robust enough to resist malicious attacks. Generally, such attacks are applied to either push/nuke popularity of specific items or damage overall performance of a recommendation system.

The concept of CF descends from the work in the area of information filtering (ACM 1992). The term "collaborative filtering" was first coined by the designers of Tapestry (Goldberg et al. 1992), a mail filtering software developed in the early nineties for the intranet at the Xerox Palo Alto Research Center. Since then the research about CF has been growing. Although shilling attack or profile injection attack concept is first introduced by O'Mahony et al. (2002a,b) in 2002, Dellarocas (2000) discusses fraudulent behaviors against reputation reporting systems. The author aims to construct more robust online reputation systems by identifying frauds. O'Mahony et al. (2002a,b) argue vulnerabilities of recommender systems against attacks to promote specific recommendations. Researchers have been studying on defining such possible attacks, detecting them, increasing robustness of recommender systems or developing robust algorithms against known attacks, and performing cost/benefit analysis. In addition, there are a number of studies compiling up-to-date developments in this field. In other words, some researchers focus on surveying about shilling attacks and their effects on recommendation systems.

Mehta and Hofmann (2008) survey about robust CF approaches only. They review some robust CF methods via intelligent neighbor selection, association rules, probabilistic latent semantic analysis (PLSA), singular value decomposition (SVD), and robust matrix factorization (RMF). They report that these approaches fall short to guarantee producing robust

recommendations under shilling attack scenarios. They also explain a relatively recent model-based approach, *VarSelect* SVD, to provide robustness to recommender systems and they show its stability to shilling. In another survey paper, Sandvig et al. (2008) examine robustness of several model-based CF techniques such as clustering, feature reduction, and association rules. Specifically, they employ principal component analysis (PCA) to calculate similarities and Apriori algorithm to produce recommendations. According to the presented results, model-based approaches are reported to be more resistive to shilling attacks than conventional nearest neighbor-based algorithms. Similarly, in the last survey paper published so far, Zhang (2009c) presents a survey of research on shilling attacks, attack detection, and attack evaluation metrics. Zhang (2009c) describes some attack models like random, average, bandwagon, segment, and reverse bandwagon attack; explains well-known attack detection approaches such as generic and model-specific attributes; and discusses prediction shift, hit ratio, and Exptop$N$ as evaluation metrics.

In addition to the abovementioned survey papers, Mobasher et al. (2007a); Mobasher et al. (2007b) categorize attack types according to their dimensions, i.e., required knowledge to realize the attack, intent of attacking, and size of attack; and then describe particular attacks with examples. Besides analyzing attack types, the authors also describe detection methods and evaluation metrics in detecting shilling attacks. They also investigate responses of model-based, hybrid, and trust-based recommender systems against shilling attacks. Burke et al. (2005a) outline some of the important issues for continuing research in robust CF systems like attack models, algorithms, profiling techniques, detection, and evaluation. Burke et al. (2011) discuss attack profiles and focus on especially some of the attack detection techniques and present some of the robust algorithms.

As stated previously, the survey conducted by Mehta and Hofmann (2008) describes robust CF approaches only. However, major researches in this field fall into four major categories: describing shilling attacks, studying shilling attack detection algorithms, developing robust algorithms or enhancing robustness, and evaluating proposed schemes. Thus, their survey focuses on one of the major trends only and leaves more works to be done. Similarly, the work conducted by Sandvig et al. (2008) focuses on robust model-based algorithms only. Hence, its scope is also limited and leaves more investigations to be performed. Zhang (2009c) discusses limited number of attack types, attack detection strategies, and evaluation metrics; thus, falls short leaving more works to be done. Furthermore, the previous survey papers introduce information about development in researches about manipulating recommender systems until 2009 only. On the other hand, there are several new works about robustness of recommender systems presented since then. Likewise, the works presented in Burke et al. (2005a); Burke et al. (2011), Mobasher et al. (2007a); Mobasher et al. (2007b) also fall short covering all related studies. Hence, in this survey, we extensively cover attack types, attack detection schemes, robust algorithms, techniques to improve robustness, cost/benefit analysis, and the new studies. We also give future directions about the field and discuss some open questions. We summarize the abovementioned surveys and current survey with respect to contents of several aspects of shilling attacks in Table 1.

## 3 Shilling attack types

In order to enhance the robustness of a recommender system against any possible attack, we need to first understand for which purpose attacks are performed and how generally they are realized. Common motivation behind almost all shilling attacks is to either push or nuke a particular product's popularity to gain economical advantage over competitors. Generally

**Table 1** Comparison of previous surveys and current study

| Comparison attributes | Survey papers | | | | |
|---|---|---|---|---|---|
| | Mehta and Hofmann (2008) | Sandvig et al. (2008) | Zhang (2009c) | Mobasher et al. (2007b) | Current survey |
| Attack types | | | | | |
| Definition | + | ++ | + | ++ | ++ |
| Profile generation | – | + | + | ++ | ++ |
| Coverage | + | + | + | ++ | ++ |
| Classification | – | – | – | + | ++ |
| Detection methods | | | | | |
| Definition | – | + | – | + | + |
| Coverage | – | + | – | + | ++ |
| Robust algorithms | | | | | |
| Definition | ++ | – | – | – | + |
| Coverage | ++ | – | – | – | ++ |
| Cost/benefit analysis | – | – | – | – | ++ |
| Statistical analysis | – | – | – | – | ++ |
| Evaluation components | | | | | |
| Data sets | – | – | ++ | ++ | ++ |
| Metrics | ++ | ++ | ++ | ++ | ++ |
| Discussion | ++ | – | – | – | + |

–, not mentioned, +, thoroughly analyzed, ++ deeply analyzed

speaking, a push attack is designed to increase the popularity of a target item so that the recommender system returns it as a recommendation to their customers. On the other hand, a nuke attack is designed to decrease the popularity of a target item so that the probability of the target item being recommended will be reduced (Mobasher et al. 2007b).

To perform any shilling attack, the attackers need to know some information about the recommender system that they try to attack. Such information might include but not limited to the mean rating and standard deviation for each item and/or user in the user-item matrix, ratings distribution, and so on. Low-knowledge attacks require system independent knowledge that might be obtained through public sources. Yet, high-knowledge attacks require very detailed knowledge about the recommender system and ratings distribution (Mobasher et al. 2007b). Compared to low- or high-knowledge attacks, informed attacks require the most information about the target CF system. It is necessary to obtain the high degree of domain knowledge that is required to select appropriate items and ratings used to generate attack profiles (Burke et al. 2011).

Attackers usually realize shilling attacks by injecting an attack profile as shown in Fig. 1, which is first defined by Bhaumik et al. (2006), Mobasher et al. (2007a) to mislead the CF system. Such profiles can be defined as four set of items (Bhaumik et al. 2006; Mobasher et al. 2007a). Initially, a set of items, $I_S$, is determined by the attacker together with a particular rating function $\delta$ to form the characteristics of the attack. Also, another set of items, $I_F$, is selected randomly with a rating function $\theta$ to obstruct detection of an attack. Finally, a unique item $i_t$ is targeted with a rating function, $\Upsilon$, to form a bias on. Remaining items

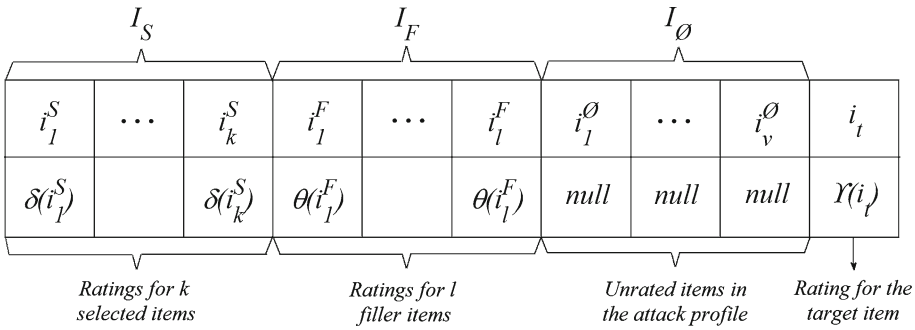


**Fig. 1** General form of an attack profile

**Table 2** Attack types according to intent and required knowledge dimensions

| Attack type | Intent | | Required knowledge | | |
|---|---|---|---|---|---|
| | Push | Nuke | Low | High | Informed |
| Random (RandomBot) | ✓ | ✓ | ✓ | | |
| Average (AverageBot) | ✓ | ✓ | | ✓ | |
| Probe | ✓ | ✓ | | | ✓ |
| Bandwagon (Popular) | ✓ | | ✓ | | |
| Segment | ✓ | | ✓ | | |
| Reverse Bandwagon | | ✓ | ✓ | | |
| Love/hate | | ✓ | ✓ | | |
| Hybrid | ✓ | ✓ | ✓ | | |
| Consistency (favorite item) | ✓ | ✓ | | ✓ | |
| Perfect knowledge | ✓ | ✓ | | ✓ | |

are left unrated indicated as $I_\varphi$ in Fig. 1. Ray and Mahanti (2009a) propose an intelligent strategy for selecting filler items to realize more effective shilling attacks. Attacking a system can be briefly defined, as follows: Malicious user or company acts as authentic users and creates fake profiles (note that user preferences about various items represented in a vector is referred to as the profile of that user). She then sends them to the recommender system to attack (she injects such fake profiles into the attacked system's database).

Typically, an attack is realized by inserting several attack profiles into a recommender system database to cause bias on selected target items. Attacks might be applied with different purposes and they can be classified according to different dimensions (Lam and Riedl 2004), i.e., intent of attack, targets, required knowledge, cost, algorithm dependence, and detectability. However, attacks are categorized mostly relying on two dimensions in the literature, i.e., intent of attack and required knowledge to apply it. The most well-known attack types are listed according to these two widely used classification dimensions in Table 2. Although some well-known attack profiles (random, average, bandwagon, segment, love/hate, and reverse bandwagon) are simply given in Mobasher et al. (2007b), we extend such list by covering other attack profiles (probe, hybrid, and consistency) and show the attack profiles of popular attack types in Table 3 according to general attack profile given in Fig. 1.

**Table 3** Attack profile summary

| Attack type | $I_S$ | | $I_F$ | | $I_\emptyset$ | $i_t$ |
|---|---|---|---|---|---|---|
| | Items | Rating | Items | Rating | | |
| Random (randomBot) | Not used | | Randomly chosen | System mean | $I - I_F$ | $r_{max}/r_{min}$ |
| Average (averageBot) | Not used | | Randomly chosen | Item mean | $I - I_F$ | $r_{max}/r_{min}$ |
| Probe | Randomly chosen | System response | Seed items | True preferences | $I - \{I_F \cup I_S\}$ | $r_{max}/r_{min}$ |
| Bandwagon (popular) | Popular items | $r_{max}$ | Randomly chosen | System mean | $I - \{I_F \cup I_S\}$ | $r_{max}$ |
| Segment | Segmented items | $r_{max}$ | Randomly chosen | $r_{min}$ | $I - \{I_F \cup I_S\}$ | $r_{max}$ |
| Reverse bandwagon | Unpopular items | $r_{max}$ | Randomly chosen | System mean | $I - \{I_F \cup I_S\}$ | $r_{min}$ |
| Love/hate | Not used | | Randomly chosen | $r_{max}$ | $I - \{I_F \cup I_S\}$ | $r_{min}$ |
| Hybrid | Popular/known average items | $r_{max}$/item average | Randomly chosen | System mean | $I - \{I_F \cup I_S\}$ | $r_{max}/r_{min}$ |
| Consistency (favorite item) | Favorite items of a user | $r_{max}$ | $I - I_S$ | Random | $\emptyset$ | $r_{max}/r_{min}$ |

As seen from Table 2, shilling attacks can be classified as push or nuke according to their intent. Likewise, they are grouped as low, high, or informed attacks with respect to required knowledge. Although some attacks can only be used for either to push or nuke an item, some can be used for both intents. As seen from Table 2, attacks generally require low knowledge. Average, consistency, and perfect knowledge attacks, on the other hand, require high knowledge. Unlike the others, probe attack is classified as informed knowledge attack because it is vital to get domain knowledge about the items and ratings with which the attacks are generated.

The most well-known attack types can be briefly described, as follows: *Random attack* operates through attack profiles with ratings to randomly chosen empty cells around system overall mean and $r_{max}$ or $r_{min}$ to target item for push and nuke attacks, respectively. Random attacks are easy to implement, however, not very much effective (Burke et al. 2005a,b; Burke et al. 2006a; Mobasher et al. 2007a; Mobasher et al. 2007b; Ray and Mahanti 2009b). This type of attack is alternatively called RandomBot attack (Chirita et al. 2005; Lam and Riedl 2004). *Average attack* operates through attack profiles with ratings to randomly chosen empty cells around each item's mean and $r_{max}$ or $r_{min}$ to target item for push and nuke attacks, respectively. This attack requires high level knowledge; and therefore, it is hard to implement (Burke et al. 2006b; Mehta et al. 2007a,b; Mehta and Nejdl 2008; Mobasher et al. 2007b; Ray and Mahanti 2009b). Alternatively, it is called AverageBot attack (Hurley et al. 2007; Lam and Riedl 2004).

In *bandwagon or popular attack*, an attacker generates profiles with high ratings to well-known popular items and the highest possible rating to the target item. This way, injected profiles can easily be associated in terms of similarity to other users in the system and push the predictions to the target item. This attack is easy to implement because it requires public knowledge rather than domain specific knowledge and as effective as the average attack (Cheng and Hurley 2010b; O'Mahony et al. 2005, 2006b). *Reverse bandwagon attack* is a variation of bandwagon attack to nuke particular products. In this attack, profiles are generated based on giving low ratings to least popular items and target item. Similarly, reverse bandwagon attack is relatively easy to implement (Mobasher et al. 2007a; Zhang 2009a).

*Probe attack* generates profiles by responses of recommender itself. Initially assigning authentic ratings to a small number of seed items, attacker queries the system and builds up attack profiles very similar to existing users in the system. This way, a target item can be biased in a neighborhood based recommender system easily. Another advantage of this attack besides being simple is that it requires less domain knowledge (Burke et al. 2005a; Hurley et al. 2007; Mobasher et al. 2007a; O'Mahony et al. 2005, 2006b). *Segment or segmented attack* is designed to target a specific group of users who are likely to buy a particular product. In attack profiles, attacker inserts high ratings for products the users in the segment probably like, and low ratings for others. This way, similarity between users in the segment and injected profiles appears high, and so target item becomes more likely to be recommended (Burke et al. 2005c,d; Hurley et al. 2007; Mobasher et al. 2007a; Sandvig et al. 2007a).

*Love/hate attack* is an extremely effective nuke attack in which randomly chosen filler items are rated with the highest possible rating while the target item is given the lowest one in attack profiles (Mobasher et al. 2007a; Zhang 2009a). *Hybrid attack* combines average and bandwagon attacks. While some of the items $I_S$ are selected from popular items others are chosen from the ones attacker knows their average rating (Zhang 2011b). *Consistency or favorite item attack* relies on actual user preferences and produces attack profiles consistent with popular or unpopular items of corresponding user for push and nuke attacks, respectively (Burke et al. 2005a,b).

In *perfect knowledge attack*, the attacker reproduces the precise details of the data distribution within the profile database or user-item matrix (Burke et al. 2005a). It focuses on reproducing fake profiles preserving the data distribution of original data precisely, except by giving high or low ratings to a particular item more frequently. However, for an attacker, it is not very realistic to be able to obtain so accurate information on data source.

Other than those mentioned attacks, there are also some rarely encountered attacks in the literature. *Sampling attack* is one such attack in which profiles are generated from samples of actual profiles by augmenting particular items' rating (Burke et al. 2005b). Although it shows unstable structure of CF approach, it is relatively impractical to realize this type of attack as original user profiles are not easily accessible (Burke et al. 2005b; Mobasher et al. 2005; O'Mahony et al. 2004c). Lastly, for reputation-based recommender systems, there is an attack type called *copied item injection attack*, which is applied by attacker to boost her reputation in the system and consequently, attract attention to her other items she owns (Oostendorp and Sami 2009).

There are also some attack strategies, referred to as the *obfuscated attacks* that an attacker might use to modify an attack to avoid attack detection (Williams et al. 2006b; Williams 2006). Williams et al. (2006b) describe three methods that can be used to obfuscate standard attack models, which can be named as obfuscated attacks. *Noise injection* is designed to mask the signature of common attack models, where random numbers generated from Gaussian distribution are added to each rating within a set of attack profile items. *User shifting* is designed to reduce the similarity between profiles of an attack. It involves shifting all ratings for a subset of items per attack profile. The goal in *target shifting* is to reduce the extreme ratings of attack profiles. In case of push attack, it involves shifting the rating given to target item from maximum to minimum. Conversely, in case of nuke attack, target item rating is shifted from minimum to maximum. Hurley et al. (2009) introduce an obfuscated attack model called the *average over popular items* (AoP) *attack*. Average attack is obfuscated by choosing filler items with equal probability from the top $X$ % of most popular items, where $X$ is selected to ensure non-detectability. Bhaumik et al. (2011) propose *mixed attack*, which involves attacking the same target item and producing from different attack models. Cheng and Hurley (2009b) propose a strategy to obfuscate attack by adding addition filler ratings in the most unpopular items, setting some these filler ratings to the maximum rating and others to the minimum rating.

As structured in Table 2, there are two major dimensions determined in the literature in order to classify shilling attacks. In addition to intent and required knowledge dimensions, three more dimensions can be introduced: application, in which recommender systems are utilized, algorithms to which the attacks are applied, and vote type. First of all, recommender systems can be employed in different applications. The most well-known application areas of them are e-commerce sites utilizing CF algorithms to provide predictions and Web-based recommender systems utilized by online reputation systems, tagging systems, page recommendation schemes, and so on. CF algorithms operate on users ratings. However, unlike pure CF schemes, Web-based recommender systems also employ additional content information to provide referrals. Due to their different nature, in order to attack such systems, application-oriented attacks should be developed.

CF algorithms are usually grouped into three major classes: memory-based, model-based, and hybrid CF algorithms. Memory-based ones operate over the entire user-item matrix to estimate predictions (Breese et al. 1998). Model-based algorithms, on the other hand, first create a model off-line from user-item matrix; they then used that model to produce predictions online (Breese et al. 1998). Although model-based CF schemes are faster than memory-based ones, their accuracy is slightly worse than memory-based ones' accuracy. Hybrid approaches

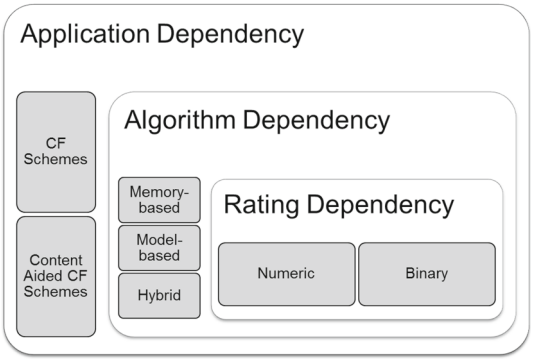


**Fig. 2** More classification dimensions for grouping shilling attacks

**Table 4** Attack types classified according to additional dimensions

| Attack type | Application | | Algorithm | | Rating | |
|---|---|---|---|---|---|---|
| | CF | Content aided CF | Memory | Model | Numeric | Binary |
| Random | ✓ | | ✓ | ✓ | ✓ | ✓ |
| Average | ✓ | | ✓ | ✓ | ✓ | ✓ |
| Probe | ✓ | | ✓ | | ✓ | |
| Bandwagon | ✓ | | ✓ | | ✓ | ✓ |
| Segment | ✓ | | ✓ | ✓ | ✓ | |
| Reverse bandwagon | ✓ | | ✓ | | ✓ | |
| Love/hate | ✓ | | ✓ | | ✓ | |
| Consistency | ✓ | | ✓ | | ✓ | |
| Perfect knowledge | ✓ | | ✓ | | ✓ | |
| Crawling attacks | | ✓ | ✓ | ✓ | ✓ | |

combine the advantages of both memory- and model-based CF algorithms (Pennock et al. 2000). Since these three different types of algorithms have different properties, in order to attack them, different shilling attack strategies should be developed.

Finally, users' preferences about various products can be represented using ratings. Such ratings can be numeric (either discrete or continuous) or binary. Most CF systems collect numeric ratings in which higher numbers represent most liked products while lower numbers show most disliked items. On the other hand, binary ratings show whether an item is liked or disliked. Since different CF systems might collect different types of ratings, attacking their user-item matrix requires different shilling attacks. Thus, we propose new classification dimensions and attributes to group shilling attacks, as shown in Fig. 2.

The proposed classification dimensions in Fig. 2 can be considered in relation to each other, e.g., an attack type can target a content aided CF scheme employing model- and memory-based hybrid algorithm operating over numerical preferences. According to new dimensions that we introduced, various attack types can be categorized, as shown in Table 4. Shilling attacks like random, average, and bandwagon attacks can be applied against both memory- and model-based shilling attacks including numerical and binary data (Cheng and Hurley 2009a; Long and Hu 2010).

As stated above, although there are various attack strategies that can be applied to memory- or model-based recommendation algorithms, to the best of our knowledge, there are no attack strategies that can be applied to hybrid CF algorithms. Attacking hybrid recommendation approaches require more knowledge, talent, and labor compared to attacking memory- or model-based schemes. The reason for this can be explained with respect to the dual nature of hybrid schemes because such algorithms are usually combination of memory- and model-based algorithms.

Shilling attack types proposed so far are usually used to attack numeric rating-based CF schemes. All push and nuke attack strategies are based on the assumption that users' preferences are represented using numeric ratings. Compared to binary ratings, numeric ratings are more widespread and most of the CF algorithms are proposed to estimate predictions from numeric ratings. However, in some cases like market basket analysis, users' preferences can be shown using binary ratings; and the customers and/or online vendors are more interested in whether an item will be liked or disliked rather than how much such an item will be liked or disliked. Thus, in addition to numeric ratings-based CF schemes, there are binary ratings-based recommendation schemes (Long and Hu 2010; Miyahara and Pazzani 2002). Due to different nature of such ratings, binary ratings-oriented shilling attacks should be developed. Long and Hu (2010) investigate binary $k$-nn CF algorithm under different types of profile injection attacks using a benchmark data set. They evaluate binary ratings-based algorithm under average, random, and bandwagon attacks, compare it with user $k$-nn algorithm, and show that it is more robust than user $k$-nn algorithm. The authors consider market basked data to find out similarities.

Shilling attacks can also be grouped as those applied to CF schemes or content-based (Web-based schemes). The proposed push or nuke attacks so far can be grouped as those applied to CF schemes. They are usually applied to memory-based CF approaches, which operate on users ratings about various products rather than the content of products. In the literature, there are some proposed attacks types applied to Web- or content-based recommendation systems. Bhaumik et al. (2007a,b) investigate *crawling attacks* against navigation-based Web recommender systems. They show that Web recommenders using implicit Web navigation profiles are also subject to manipulation. They introduce *popular page* and *localized page attacks*, which happen to be practical and effective attack types. O'Mahony and Smyth (2007b) consider attack scenarios whereby malicious agents seek to promote particular result pages within a community. They propose *promote-always attack*, which seeks to promote a particular target page for all searches. In addition, they introduce *term-specific attack* whose objective is to promote a particular target page when a specific term is used in search queries. They show that the key characteristics of communities have implications for robustness.

## 4 Major research fields

Studies about profile injection attacks against CF schemes can be grouped into four major classes. The first group consists of studies focusing on shilling attack strategies and generating profile injection attacks. The second group of researchers focuses on shilling attack detection algorithms. The third group includes studies about robust CF algorithms against shilling attacks. The studies fall into this group can be further grouped as analyzing robustness of various algorithms with respect to shilling attacks and those proposing robust algorithms. The last group of studies performs cost/benefit analysis of shilling attacks and discusses cost of mounting profile injection attacks.

### 4.1 Shilling attacks and their strategies

Initially, attack strategies towards existing CF systems are discussed by O'Mahony (2004) in his dissertation. The attacks analyzed in O'Mahony (2004) are implemented by inserting bogus data through the normal system interface and no other access to a system's database is assumed. It is shown that some statistical knowledge on database is sufficient to direct attacks against recommender systems. Then in O'Mahony et al. (2005), extent of such knowledge is analyzed and it is presented that if even little such knowledge is known, effective attacks can be mounted on recommender systems. Later, Lam and Riedl (2004) discuss four open questions related to the effectiveness of proposed shilling attack types, i.e., utilized recommender system algorithm, whether recommendation or prediction is generated, detectability of attacks by system operators, and properties of items being attacked. In addition, the same authors discuss attack effectiveness, difficulty, and detection concepts in general (Lam and Riedl 2005). Also, Lam et al. (2006) extend their previous work by further discussing open researches about possible attacks on privacy-preserving prediction schemes.

After attack strategies are determined, several attack types are developed. Mobasher et al. (2007a); Mobasher et al. (2007b) discuss particular attack types such as random, average, bandwagon, and love/hate attacks; and study their impact on algorithm-specific environments. Additionally, Burke et al. (2005b) focus on attacks on user-based prediction schemes requiring limited knowledge and examine bandwagon and favorite item attacks in detail. Similarly, Burke et al. (2005c,d) and Mobasher et al. (2006a) introduce segment or segmented attack model concentrating on a targeted set of users. They present the effectiveness of this attack model both against user- and item-based systems. Bhaumik et al. (2007a) introduce crawling attacks, which require inside access to a recommender system and based on clickstreams of user profiles. They give several examples of such crawling attacks (Bhaumik et al. 2007a) and also analyze effectiveness of those attacks against some web personalization algorithms (Bhaumik et al. 2007b). Besides memory-based algorithms, model-based approaches are also subject to shilling attacks. Cheng and Hurley (2009b) study how an effective attack can be mounted on model-based algorithms and they propose diverse and obfuscated attack models to be effective on such schemes. The same authors explore informed model-based attacks and analyze trade-off between privacy and robustness in peer-to-peer (P2P) recommender systems (Cheng and Hurley 2009c). Oostendorp and Sami (2009) define an attack type called copied-item injection attack to bias user reputation in those systems. Lastly, Ramezani et al. (2009) develop a framework for characterizing various shilling attacks against tagging systems. Lang et al. (2010) examine social network-based recommender systems and they show that users in such networks can successfully manipulate other users' recommendations.

### 4.2 Shilling attack detection

Detecting attack profiles is important to reduce effects of known attack types and protect favoring of targeted items. For this purpose, researchers propose several techniques, which can mainly be thought as statistical, clustering, classification, and data reduction-based methods.

#### *4.2.1 Statistical techniques*

Researchers propose statistical analysis methods to detect anomalies in databases caused by suspicious ratings. *Statistical anomaly detection* (Bhaumik et al. 2006) is one such approach relying on item average values. Outlier items are determined using two statistical process

control techniques, i.e., X-bar control limit and confidence interval control limit. In the former one, how far item averages are from the overall mean are measured and three-sigma standard deviation from the overall mean is considered safe zone for items. In the latter, a confidence interval is determined at a confidence rate assuming the population has a normal distribution. Items having an average falling outside of the confidence interval are considered suspicious. Similarly, Hurley et al. (2009) utilize Neyman-Pearson *statistical detection theory* in which a binary hypothesis testing is performed to discriminate between genuine and attacker profiles. In addition, Li and Luo (2011) propose to utilize *probabilistic Bayesian network models* to test whether a new profile is malicious or authentic. Similarly, Zhang et al. (2006a) propose a *probabilistic approach* using SVD-based data reduction method, where a compacted model of observed ratings (including real and biased ones) is generated maximizing the log-likelihood of all ratings. Then the degree of belief in a rating can be estimated as log-likelihood of this rating given the compacted model.

#### *4.2.2 Classification*

*Supervised classification* techniques are utilized in attack detection schemes. Burke et al. (2006a,b) utilize a classification approach for detecting malicious users based on attributes derived from each individual profile. These are called *generic attributes* in literature. Two of such derived attributes are *rating deviation from mean agreement* (RDMA) and *degree of similarity with top neighbors* (DegSim) proposed by Chirita et al. (2005) as a metric for detecting malicious profiles. RDMA examines the profile's average deviation for each item weighted by the inverse of the number of ratings for corresponding item. DegSim is estimated as the average of similarities of a user and her nearest neighbors. Burke et al. (2006a,b), alternatively, consider these metrics as a classification attribute and derive two new attributes relying on RDMA. These reproduced attributes are *weighted deviation from mean agreement* (WDMA) and *weighted degree of agreement* (WDA). WDMA is strongly based on RDMA; however, it places a higher weight for sparse items by squaring the number of ratings per item. WDA, on the other hand, completely ignores number of ratings per item and utilizes sum of the differences of the user's ratings from the item average. In addition to RDMA based attributes, one more generic attribute is also proposed in Burke et al. (2006a), called *length variance* (lengthVar), which measures how much the length of a given user profile varies from the average length in the database, where length is the number of ratings of a user.

Studies on detecting attacks figured out that classification schemes relying solely on generic attributes fall short to distinguish malicious profiles from especially eccentric authentic profiles. Such incidences particularly happen when profiles are small and contain fewer filler items are small. For such circumstances, Burke et al. (2006a) propose to employ *attack model specific attributes*. This type of detection model aims at discovering partitions of profiles maximizing their similarity to a particular attack model. Partitioning is performed by dividing each user profile into two sets containing all items in the profile with the profile's maximum (or minimum for nuke attacks) rating ($P_{u,T}$) and the set of remaining filler items ($P_{u,F}$). However, this assumption is made by the server as $P_{u,T}$ to estimate $I_S \cup i_t$ and $P_{u,F}$ to $I_F$, where note that $I_S$ is the set of items determined by the attacker together with a particular rating function $\delta$ and $i_t$ is a unique item targeted with a rating function $\gamma$ to form a bias on. Statistical features of these partitions are utilized as classification attributes to detect several attack models. For example, since average attacks insert ratings to items in $P_{u,F}$ close to mean of each filler item, it is expected that the variance between a filler item's rating and that item's mean rating is lower in attack profiles. Therefore, mean variances of items in

$P_{u,F}$ can be calculated and utilized as a classification attribute for average attack detection (Burke et al. 2006a; Williams 2006). Conversely, a low correlation is expected between the filler items and the individual ratings because ratings for filler items are chosen randomly around overall system average in random attack (Williams 2006). Unlike the *average* and *random model-specific attributes*, which are designed to discover characteristics in partitions, *segment attack model-specific attributes* focus on capturing differences between partitions. Therefore, the idea for detecting segment attacks is to utilize a partitioning feature as the difference in ratings of items in $P_{u,T}$ compared to the items in $P_{u,F}$ (Burke et al. 2006a; Williams 2006).

Proposed classification attributes thus far focus on discovering characteristics in a single profile. However, if an attack is performed on a target item, it is most likely that more than one attack profiles are inserted to produce a significant bias on a particular item. Therefore, another class of attribute is introduced focusing on statistics across profiles called *intra-profile detection attributes* (Mobasher et al. 2007b). These attributes utilize the outputs of partitioning identified by model-specific attributes to reveal attention on target items. To this end, these attributes measure the degree for which the partitioning of a given profile focuses on items common to other attack partitions.

Studies utilizing mentioned classification attributes employ commonly *k*NN, C4.5, and SVM classifiers to detect attacks (Burke et al. 2006a,b; Mobasher et al. 2006a; Mobasher et al. 2007b; Williams 2006; Williams et al. 2006a, 2007a). Alternatively, Mobasher et al. (2007b) propose a *hybrid attack detection model* employing both statistical techniques and classification using model-specific attributes. Like Mobasher et al. (2007b), Wu et al. (2011) also propose a *hybrid semi-supervised approach* combining naïve Bayesian classifiers and augmented expectation maximization. Also, He et al. (2010) exploit *rough set theory* to perform a classification approach as labeling user profiles as either being attacker or authentic.

#### *4.2.3 Unsupervised clustering*

Clustering approach in attack detection is first utilized by O'Mahony et al. (2003) as neighborhood selection is filtered through clustering to eliminate suspicious users. They utilize the *clustering approach* used in reputation reporting systems with some modifications to detect malicious profiles aiming to nuke targeted items' popularity. This approach clusters the database periodically and checks if cluster centers are changing significantly. If so, extreme profiles affecting cluster centers are considered malicious. Mehta (2007) and Mehta and Nejdl (2009) introduce a *PLSA-based clustering method* for clustering users to determine users to be utilized in recommendation generation process instead of using traditional nearest neighbor method. Bhaumik et al. (2011) apply *an unsupervised clustering algorithm* based on several classification attributes for attack detection relying on statistical characteristics of data set and accordingly produce user profiles relying on those attributes. They apply *k*-means clustering on produced profiles to locate relatively small clusters as suspicious user groups.

#### *4.2.4 Variable selection*

Although all shilling profiles resemble high similarity to each other and can form a cluster, it is not that easy to group those profiles using clustering techniques, because they also can be very similar to genuine users, as well. For this purpose, *variable selection* (to select users) is applied as an alternative method to discriminate between malicious and genuine users.

Mehta et al. (2007a) propose applying *PCA-based variable selection* by computing covariance between users to locate spam users in a recommender system. Normally, PCA interests on selecting dimensions providing most information. However, Mehta et al. (2007a) propose to select least information providing dimensions because they resemble low covariance with other users; thus, being suspicious of being spam. Similarly, Mehta (2007) and Mehta and Nejdl (2009) utilize this approach for detecting attack profiles.

#### *4.2.5 Other detection techniques*

In addition to mentioned approaches, researchers also propose some other techniques to detect attacks in recommender systems. O'Mahony et al. (2006a) propose a *signal detection approach* to discover natural and malicious noise patterns in a recommender system database by estimating and interpreting probability distributions of user profiles. Su et al. (2005) focus on detecting group shilling users rather than individual attackers by employing a *similarity spreading algorithm* in which the consistency of ratings of a user is tested according to item-item similarities. Tang and Tang (2011) analyze rating *time intervals* of users to detect suspicious behavior to bias top-$N$ lists in recommender systems. They generate an attribute to measure time behavior of users consisting of span, frequency, and mount properties. As a similar approach, Zhang et al. (2006b) propose to construct a time series of ratings for an item. They use a window to group consecutive ratings for the item, compute the sample average and entropy in each window, and interpret results to detect suspicious behavior. Zhang (2011a) focuses on preventing trust-based recommender systems from attacks using a *data lineage method* to trace recommendation history and locate victim nodes. Similarly, Zhang et al. (2009) propose modifying RDMA attribute (TWDMA) weighted by trust values to be utilized as a metric to prevent trust-based systems. Bryan et al. (2008) employ a sparse matrix variation of the $H_v$*-score metric* incorporating additional steps to maximize retrieval of attack profiles.

### 4.3 Robustness analysis

O'Mahony et al. (2002a,b) introduce *robustness* as a new performance metric to compare recommendation algorithms. The authors investigate memory-based CF algorithm under push and nuke attacks in terms of robustness using benchmark data sets. They show that shilling attacks can be devised to degrade performance of CF system. In O'Mahony et al. (2004c), the authors scrutinize the ability of a CF system to make predictions despite noisy ratings. The authors define robustness in terms of *accuracy* and *stability*. They present a framework to analyze recommendation stability in the context of memory-based algorithm using real data sets. Their analysis demonstrates that CF schemes are vulnerable to attack. O'Mahony et al. (2004d) examine the effects of two neighborhood formation schemes ($k$-nn and *similarity thresholding*) and various similarity measures on robustness of memory-based schemes. Mobasher et al. (2006b) study the robustness of model-based CF schemes with respect to shilling attacks because memory-based algorithms have been shown to be vulnerable to profile injection attacks. Their empirical analysis based on two model-based algorithms (one based on $k$-means clustering and the other based on PLSA) shows that the robustness of such algorithms is better than the robustness of traditional memory-based ones.

O'Donovan and Smyth (2006), on the other hand, discuss the robustness of trust-based recommendation approach. They propose five strategies for building the trust model. Their empirical analysis under average attack shows that trust models can either reduce or increase prediction shift depending on the model building process. Williams et al. (2006a, 2007b)

examine the integrated effectiveness of various attributes for attack detection at improving the robustness of memory-based CF systems. They show that *classifiers* built using such attributes can improve the robustness. Zhang et al. (2006a) study the effectiveness of random and average attacks on a low-dimensional linear model generated using SVD for prediction estimation. According to their empirical results, SVD-based CF scheme is much more resistant to random and average attacks than memory-based algorithms. Mobasher et al. (2007b) examine user- and item-based CF systems under various attacks. Although item-based approaches are generally considered robust against attacks, the authors show that item-based scheme is vulnerable to segment attack. They also show that combination of CF schemes with other types of recommendation components provides defensive advantages in the face of profile injection attacks.

O'Mahony and Smyth (2007a,b) examine collaborative web search in terms of robustness. They consider malicious agents that aim to promote specific pages within a community. Their robustness analysis shows that community homogeneity has implications for system robustness. Sandvig et al. (2007a) compare the CF algorithm based on association rules with $k$-nn and two model-based algorithms (using $k$-means clustering and PLSA) with respect to robustness against shilling attacks. The empirical outcomes show that the scheme on association rules offer improvement in robustness. In another study (Sandvig et al. 2007b), the same authors investigate whether *relevance measures* like *significance* and *trust weighting* improve robustness or not. They integrate such relevance measures into neighbor selection process and show that significance weighting improves robustness. In Sandvig et al. (2008), the authors compare the standard user-based CF scheme with the model-based ones using $k$-means clustering, PLSA, PCA, and association rule mining in terms of robustness against push attacks like average and segment attacks using different parameters. Real data-based experiments show that the model-based schemes perform better than the $k$-nn approach with respect to robustness.

Zhang and Xu (2007) asses the robustness of their topic-level trust-based CF scheme under random and average shilling attacks using real data-based experiments. They compare their scheme with the standard $k$-nn approach and conclude that their scheme offers significant improvements in robustness. In the studies conducted by Zhang (2008, 2009a,b, 2010, 2011b), the author examine the topic-level trust-based recommendation algorithm with respect to robustness under reverse bandwagon, segment, average, love/hate, and bandwagon and average hybrid attack, respectively. In such studies, the author compares the topic-level trust-based recommendation algorithm with the user-based ($k$-nn) approach and experimentally shows that it performs better than $k$-nn in terms of robustness under various attacks.

Yan and Van Roy (2009) evaluate so called linear CF scheme with respect to robustness and show that such algorithm is more robust against shilling attacks than common nearest neighbor algorithms. In Yan (2009), the author provides both theoretical and empirical outcomes in order to compare the standard $k$-nn approaches with so called *linear* and *asymptotically linear* CF algorithms under shilling attacks. The results demonstrate that compared to correlation-based schemes, linear and asymptotically linear CF algorithms are more robust.

Cheng and Hurley (2009a) examine model-based schemes with respect to robustness against profile injection attacks. They focus on model-based approaches using factor analysis and $k$-means clustering. They find out that deliberate shilling attacks can be designed to make such model-based approaches vulnerable to shilling attacks as memory-based ones. In another study (Cheng and Hurley 2009c), the authors scrutinize the trade-off between robustness and privacy in P2P CF systems. They argue that divulging some information about model parameters can help attackers design more effective attacks against model-based schemes. Their analysis shows that privacy and robustness in P2P schemes conflict each other and

it becomes difficult to achieve both goals at the same time. Although trust-based schemes are considered more robust against shilling attacks, Cheng and Hurley (2010a) argue that revealing ratings in trust-based schemes can cause robustness vulnerability in such systems. They also show that such vulnerability can be exploited to design more effective profile injection attacks against trust-based recommendation algorithms. Thus, the authors state that trust models should be chosen carefully.

Long and Hu (2010) assess the binary CF (in which user-user similarities are estimated based on who-rated-what rather than actual ratings) under random, average, and bandwagon shilling attacks using a real data set. Their empirical outcomes show that binary CF performs better than $k$-nn approaches with respect to robustness against shilling attacks.

### 4.4 Robust algorithms

In this subsection, we briefly discuss the studies proposing robust approaches against shilling attacks for CF. O'Mahony et al. (2003) propose a *neighborhood filtering* scheme as a robust method. The method can be basically described, as follows: First, a filtering is performed, which places the ratings into two groups via clustering. It is then assumed that the group with the highest mean contains malicious ratings. Finally, prediction is estimated from those ratings only in the genuine group. Similar neighborhood filtering scheme is proposed by O'Mahony (2004), where neighbors of an active user is first determined. Such neighbors are then clustered into two groups based on the ratings for target item. Next, the group with the smallest standard deviation across the item ratings is discarded. Finally, prediction is computed using non-bogus ratings. O'Mahony (2004) and O'Mahony et al. (2004a,b) propose an *intelligent neighborhood formation* on *profile utility* and *similarity weight transformation*. The authors define profile utility with respect to item popularity and rate utility as the average inverse popularity of all items within a profile. Profiles containing less popular items are thus assigned a higher weight. Moreover, they also transform similarity weights using the inverse popularity so that attack profiles are unlikely to significantly influence predictions.

Ji et al. (2007) propose *trust-based collaborative filtering with mobile agents* (TCFMA) method, which is a distributed scheme in P2P network based on web of trust. The method can be used to provide trustworthy predictions to users. In order to achieve shilling attack resistance, mobile agents are added to the architecture. Real data-based trials show that the scheme offers significant advantages with respect to handling shilling attacks. Trust-aware recommender systems are more robust prediction schemes than traditional ones against shilling attacks (Massa and Avesani 2007; Ray and Mahanti 2010).

Mehta et al. (2007b) propose *RMF* on *M-estimators* algorithm as a resistant scheme against shilling attacks. The authors find out that the M-estimators do not improve robustness significantly. However, the new scheme performs better than the existing CF algorithms in terms of prediction accuracy.

Mobasher et al. (2007b) show that both user- and item-based CF algorithms are highly vulnerable to specific type of attacks; however, hybrid CF schemes may improve robustness. Their hybrid approach is known as *semantically-enhanced collaborative filtering*. In their proposed scheme, a weighted hybrid approach is used, where predictions of multiple components are combined into a single score using a weighted sum. The design is a hybrid method combining item- and knowledge-based recommendation schemes.

Resnick and Sami (2007, 2008a) propose to utilize the *influence limiter* algorithm, which is a manipulation resistant CF system. The authors describe an influence-limiting algorithm that can transform vulnerable CF schemes into robust ones. In such algorithm, there are two additional features. In the first one, recommendation algorithm passes through an

influence-limiting process. The second one is a scoring function, which assigns reputation to the target item. The amount of reputation is limited by the current reputation.

In order to enhance robustness, Zhang and Xu (2007) recommend utilizing *topic-level trust-based prediction algorithm*. First of all, degree of trust is computed for a user based on its neighbors. Then, the trust is described as an item-level trust. Topic-level trust then can be estimated by calculating the average score of item-level trust for a producer on the items belonging to the same topic and have been rated by the producer. Finally, trust is incorporated into traditional CF scheme for producing predictions.

Mehta and Hofmann (2008) and Mehta and Nejdl (2008, 2009) propose *VarSelect SVD* as a robust CF method. The algorithm consists of two major steps: detection followed by removal of profiles and/or votes and model building. In order to find suspected attack profiles, VarSelect SVD uses variable selection using a selection criterion called normalized loading combination. The recommendation model is based on SVD. The user-item matrix is factorized into factors, which are then used for prediction estimation.

*Linear* and *asymptotically linear collaborative filtering algorithms* are proposed as robust recommendation algorithms by Van Roy and Yan (2009, 2010), Yan and Van Roy (2009), and Yan (2009). The authors find out that as a user votes an increasing number of items, average accuracy of recommendations made by a linear CF algorithm become insensitive to manipulated data. A linear recommendation method offers predictions based on a probability distribution, which is a combination of two distributions. The first one learns given data generated by honest users only. On the other hand, the second one learns given manipulated data only. If a user whose ratings are supposed to be predicted provides more ratings, then it becomes clear which of these two distributions better represents her preferences.

In order to enhance the robustness of the least squares-based matrix factorization CF schemes, Cheng and Hurley (2010b) propose a *least trimmed squares-based matrix factorization* (LTSMF) recommendation method. Least trimmed squares estimators are widely used robust model. The method can be briefly described, as follows (Cheng and Hurley 2010b): (1) conduct stochastic descent to initialize parameters $x$ and $y$ and record residuals, (2) order residuals and choose the smallest ones, (3) continue stochastic gradient descent on selected residuals and update $x$ and $y$, (4) calculate residuals over entire ratings and repeat from step *ii*, and (5) return $x$ and $y$ for estimating predictions.

In addition to the abovementioned robust schemes, Mobasher et al. (2006b) suggest applying *PLSA* to CF for robustness. The authors indicate that PLSA is a very robust CF algorithm and it is stable in the face of shilling attacks. Sandvig et al. (2007a) propose to apply a popular approach for frequent-pattern mining (association rules on Apriori) to CF for enhancing robustness. The authors apply Apriori to user ratings and design a robust CF scheme.

### 4.5 Cost/benefit analysis

Researchers also study the cost of mounting an attack and benefits obtained by an attacker after the attack. O'Mahony et al. (2006b) perform a cost-benefit analysis, which shows that substantial profits can be obtained by the attackers. They propose a framework to quantify the financial gains by the attackers. Their analysis is based on the *return-on-investment* (ROI) metric. The goal for an attacker is to maximize ROI with respect to the item being attacked. Their analysis based on popular attack shows that single-profile attacks with range in size from 7 to 25, ROI is the greatest. For larger attack sizes, ROI significantly decreases. Hurley et al. (2007) compute ROI as the increase in profit due to an attack, where it is determined from the increase in the number of product sales. However, calculating this requires much information about buyer practices. They also evaluate the cost-benefit analysis in the context

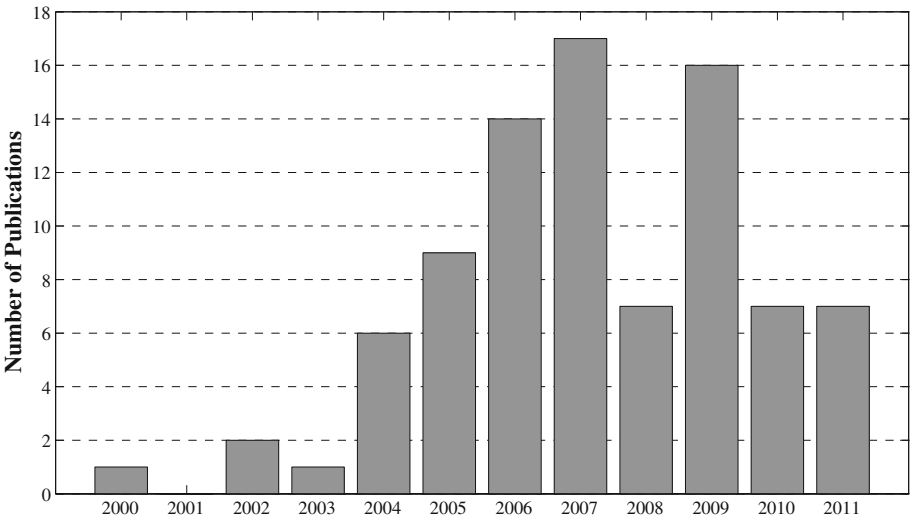


**Fig. 3** Number of publications over years

of popular attack. They compare attack costs against increase in sales, where a unit attack cost for each item is assumed. Once the expected increase in sales exceeds the attack cost, the ROI becomes positive. Similar analysis based on the ROI is also conducted in O'Mahony (2004). Oostendorp and Sami (2009) state that the cost and benefit to an attacker in executing a copied-item injection attack can be described with respect to content space model, which is based on distance between items. O'Mahony et al. (2004b) experimentally observe that cost/benefit ratio begins to fall as attack strength increase, where attack strength is defined as the number of inserted attack profiles. Unlike studies presented in O'Mahony et al. (2006b), Hurley et al. (2007), O'Mahony (2004), where shilling attacks are investigated with respect to gains versus attack costs; Resnick and Sami (2008b) examine robust CF approaches. They hypothesize that robust schemes exhibit a fundamental trade-off between robustness and optimal use of genuine ratings. They prove that any robust CF system must also discard some units of useful information from each genuine rater. In other words, some amount of information loss is inevitable in a CF scheme that resists manipulation.

O'Mahony et al. (2004c) and Chirita et al. (2005) define cost of an attack as a function of the number of attack profiles added to the user-item matrix. In Tang and Tang (2011), Mobasher et al. (2005, 2007b), Williams (2006), Williams et al. (2007a), attack cost includes *knowledge cost* (necessary information about CF systems and users' ratings patterns) and *execution cost* (effort for interacting with CF system to submit an attack). Similarly, according to Lam and Riedl (2004), attack cost depends on the amount of effort and information needed to successfully execute the attack. Attack size, difficulty of interacting with the CF scheme, obtaining required knowledge for successful attack, and other required resources contribute to the cost of the attack.

### 4.6 Statistical analysis of the studies

In this section, we summarize the abovementioned studies by giving some statistical figures. We first present the number of related publications over the years in Fig. 3.

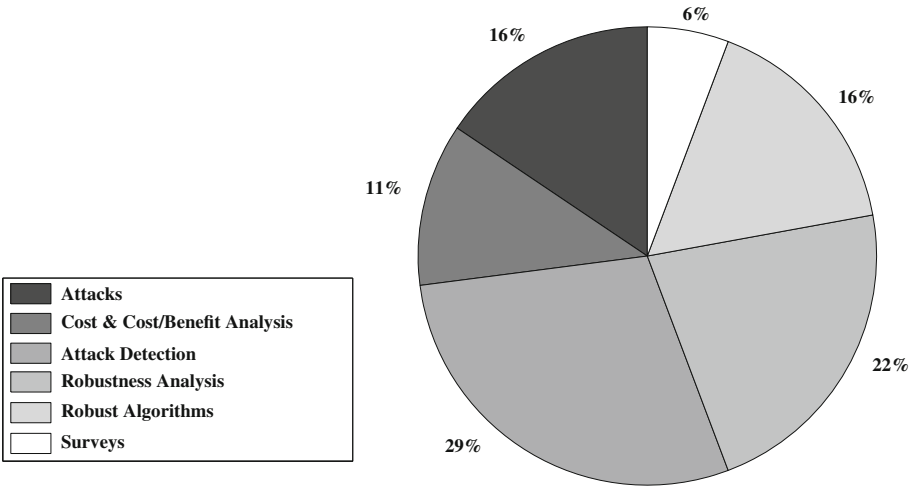


**Fig. 4** Percentages of major research directions

As stated before, the studies about shilling attacks are inspired by the work conducted by Dellarocas (2000). Number of studies increases from the year 2003 to 2007, as seen from the Fig. 3. Although number of studies seems decreasing after the year 2007, there are still significant numbers of studies. Also, new studies can be conducted to fill the missing gap in this field because shilling attacks will be in place as the popularity of recommender systems continue to grow.

We also perform a simple statistical analysis about the studies in terms of the major research directions. As stated before, there are four major research groups; and one of them can be further into two groups. Additionally, there are some studies, which are considered surveys about shilling attacks. Therefore, we group the related studies into six groups and show the outcomes as percentages in Fig. 4. As seen from Fig. 4, 26 % of the studies focus on detection algorithms. The next popular research subject is robustness analysis, which is the 22 % of all studies. Investigating attack strategies and proposing robust algorithms come next. Compared to others, cost/benefit analysis and surveys take less attention of the researchers.

After examining the studies in terms of the major areas, we analyze each group of studies over the years, where we omit surveys because their distribution over the years is discussed before; and display the outcomes in Fig. 5.

As seen from Fig. 5, studies about shilling attacks in general and developing attack strategies make peak in 2005. Developing attack detection strategies takes the attention of the researchers most in 2007 and 2009. In 2007, robustness analysis becomes the most popular over the years. Most of the robust algorithms are proposed in 2006. Similarly, most of the cost/benefit analysis is conducted in 2004. In Fig. 5, note that we consider some of the studies focusing more than one research area.

We also examine the studies with respect to where they are published. Hence, we classified them according to their publishing type and display the results as percentages in Fig. 6. As seen from Fig. 6, more than half of the studies are published in conferences. One quarter of them, on the other hand, are published in journals. 15 % of them are workshop papers, while 3 % of them are dissertation and 1 % of them are book chapters.

Furthermore, for researchers who are willing to work in shilling attacks research field, it is beneficial to have a general idea about major researchers who are actively working on this

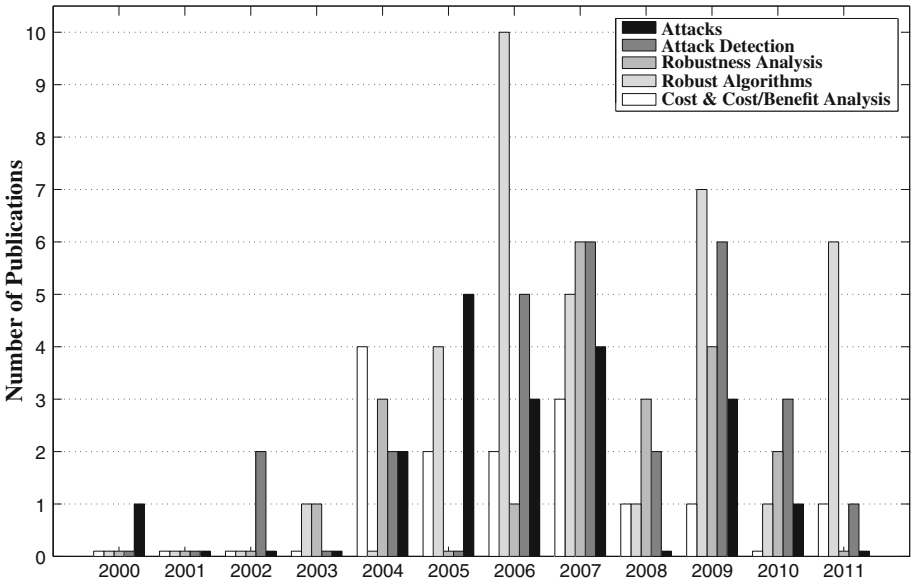


**Fig. 5** Distribution of major research directions over the years

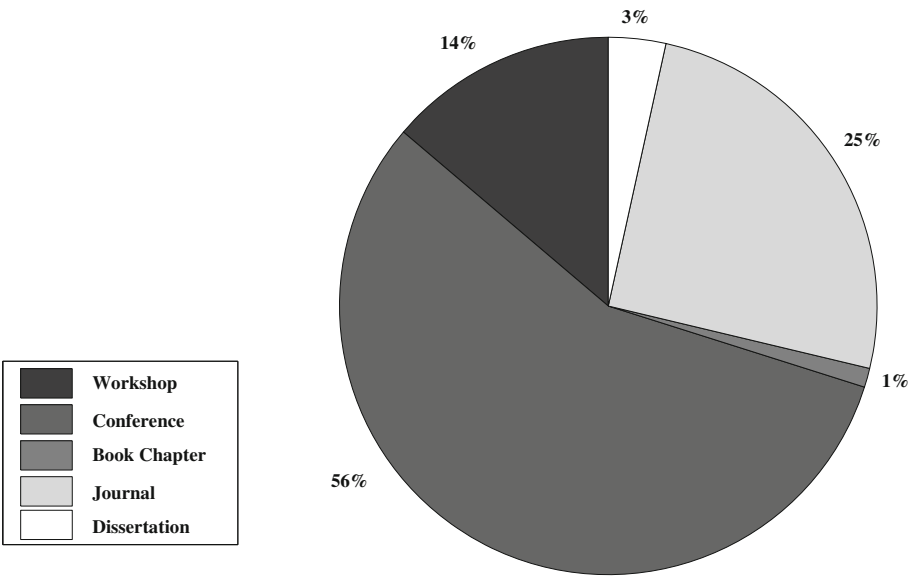


**Fig. 6** Distribution of the studies according to publishing type

field. In order to give a better understanding of the community of researchers, we present authors and their number of publications in Fig. 7. Note that, authors in Fig. 7 have at least five publications in the field.

We finally show the milestones over the years in terms of the major research areas in Fig. 8. Starting from the year 2000, we display the major studies in each area over the last decade.

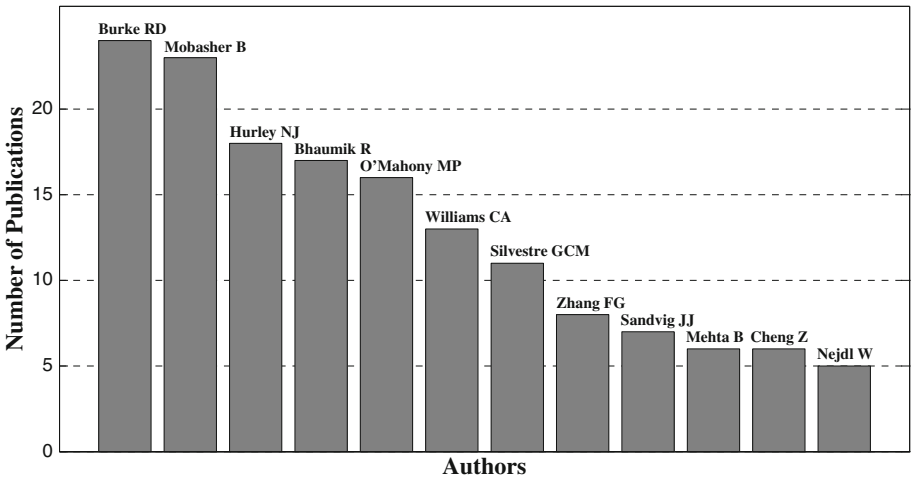


**Fig. 7** Major researchers in shilling attacks research field

These studies' outcomes can be summarized in a nutshell, as follows: After Dellarocas (2000) has initiated the idea of shilling recommender systems, O'Mahony (2004) develops basic attacks strategies. Then, in addition to developing new attack strategies, Lam and Riedl (2004) also define attack dimensions to produce practical attacks. Burke et al. (2005b) introduces limited knowledge attacks and manipulating methodologies to item-based recommendation algorithms. Although Mobasher et al. (2006b) conclude that model-based recommendation algorithms are robust against profile injection attacks, Cheng and Hurley (2009b) develop effective attacking methods for model-based algorithms. Mehta et al. (2007b) utilize matrix factorization methods to provide robustness.

In order to detect shilling profiles, Chirita et al. (2005) identify classification metrics and Bhaumik et al. (2006) introduce employing statistical anomaly detection techniques. O'Mahony (2004) defines robustness for collaborative recommendations and proposes robust algorithms to prevent shilling attacks. Ji et al. (2007) utilize web of trust in distributed recommendation schemes for protecting the recommender system against profile injection attacks. Resnick and Sami (2008b) perform a detailed analysis about cost of increases resistance of a collaborative recommender system against profile injection attacks.

## 5 Evaluation components

As discussed previously, some researchers focus on detecting shilling attacks and propose some detection algorithms. In order to show how effective their proposed approaches, they evaluate them with respect to detection accuracy using some benchmark data sets collected for CF purposes. Some researchers study how to enhance the robustness of CF schemes or they develop robust CF schemes against shilling attacks. Likewise, they scrutinize how accurate their schemes are using real data sets. Another group of researchers study the cost-benefit analysis of attacking CF schemes. Finally, some of them evaluate the effects of shilling attacks on recommender systems' accuracy.

In this section, we study evaluation metrics and benchmark data sets, which are utilized to assess the shilling attacks, detection algorithms, and robust algorithms. We first start with

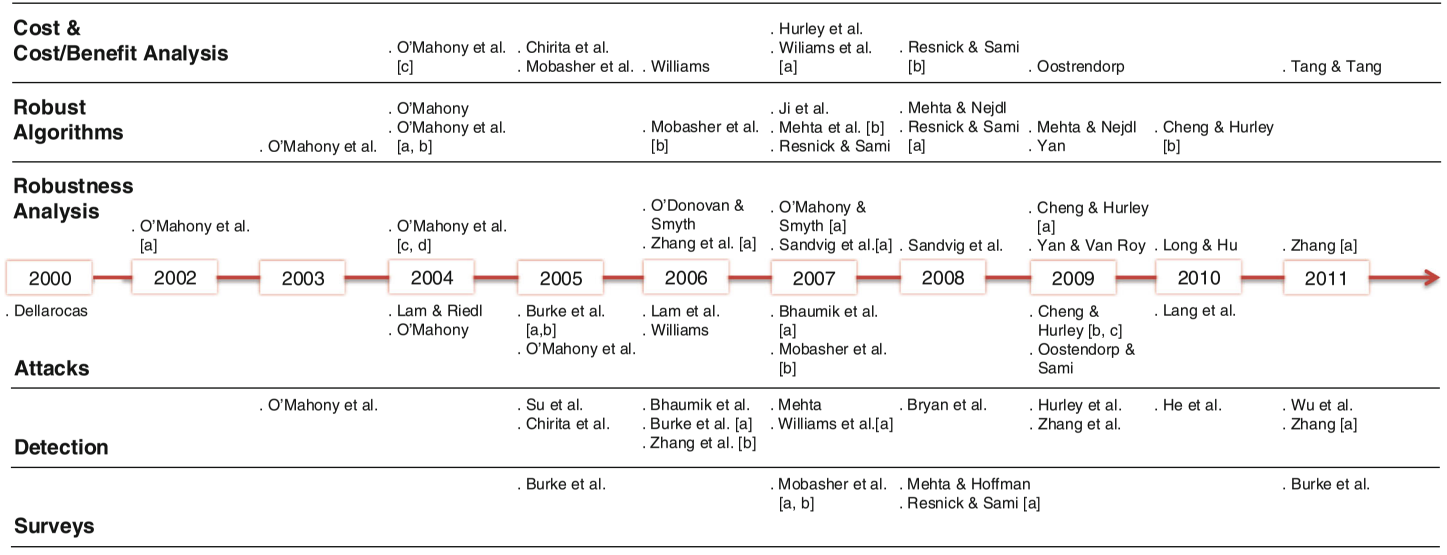


**Fig. 8** A timeline of cornerstone studies in shilling attacks

**Table 5** Data sets

| Name | Item | User × item | Range | Type | Total votes |
|---|---|---|---|---|---|
| MovieLens 100K | Movie | 943 × 1,682 | [1, 5] | 5-star | 100,000 |
| MovieLens 1M | Movie | 6,040 × 3,952 | [1, 5] | 5-star | 1,000,000 |
| EachMovie | Movie | 72,916 × 1,628 | [0, 1] | 6-star | 2,811,983 |
| Netflix | Movie | 480,189 × 17,770 | [1, 5] | 5-star | 100,480,507 |
| SmartRadio | Song | 63 × 1,055 | [1, 5] | 5-star | 4,075 |
| PTV | TV program | 2,344 × 8,199 | [1, 4] | 4-star | 60,000 |

real data sets collected for CF purposes and used for evaluation. The most commonly used data sets are described in Table 5, where the data sets are displayed in the order of their common usage.

In addition to the data sets listed in Table 5, there are less frequently used data sets like Epinions, CTI, NC, I-SPY, Mushroom, and MovieLens 10M data sets. MovieLens data sets (100K, 1M, and 10M) are probably the mostly widely used data sets collected by GroupLens (http://movielens.umn.edu/). Netflix (http://www.netflixprize.com/) and EachMovie data sets also include discrete ratings for movies. However, EachMovie data set was shut down and it not available anymore (http://www.grouplens.org/node/76). Netflix data set was published for "Netflix Prize" competition to determine the best CF algorithm. These data sets are typically very sparse due to large number of items (around 2–5 % density). SmartRadio (O'Mahony 2004; O'Mahony et al. 2004a,b) data set is obtained from the Smart Radio system, which is a personalized music recommender service. The data set is provided by the Department of Computer Science, Trinity College Dublin. PTV (O'Mahony et al. 2002a,b, 2004c) is a collaborative recommendation data for television listing (www.changingworlds.com) whose density is about 0.3 % only. Epinions (Cheng and Hurley 2010a; Ji et al. 2007) data is collected from Epinions.com, which is an online community. The users can review various items and rate them on a scale of 1 to 5. The data set includes ratings of 49,290 users for 139,738 items. CTI and NC (Bhaumik et al. 2007a,b) data sets are used for evaluating crawling attacks. CTI (Bhaumik et al. 2007a) data is based on the server logs of the host computer science department spanning one-month period. NC (Bhaumik et al. 2007a) data is based on the server logs for Network Chicago, which combines programs and activities of the Chicago Public Television and Radio (www.networkchicago.com). I-SPY (O'Mahony and Smyth 2007b) data set was obtained from a trial of I-SPY system, which took place over an extended period among the 50 staff members of a local software company. The Mushroom (O'Mahony et al. 2004c) data set from the UCI repository includes 8,124 instances with 23 attributes with no missing values.

Although various data sets are used for evaluating shilling attacks, there are other data sets collected for CF purposes, as well. They are also commonly used data sets in CF schemes. Similar evaluations can be done using additional data sets like Jester, BookCrossing, and Dating Agency. Jester is a joke recommender system (http://eigentaste.berkeley.edu/dataset/). In this data set, continuous ratings range over [-10, 10] and it is relatively much more dense (around 56 %) data set. BookCrossing contains implicit as well as explicit 10-star ratings from Book-Crossing community (http://www.bookcrossing.com/). Dating Agency contains explicit ratings for user profiles from other users of the LibimSeTi dating system (http://www.occamslab.com/petricek/data/). BookCrossing and Dating Agency data sets are extremely sparse.

**Table 6** Accuracy metrics

| Evaluating detection algorithms | Assessing shilling effects | Evaluating robust algorithms |
|---|---|---|
| Precision | Prediction Shift | MAE |
| Recall | Absolute prediction shift | NMAE |
| F1 measure | Hit ratio | RMSE |
| Correctness | High rating ratio | Coverage |
| ROC | Exptop$N$ | |

Being robust against shilling attacks has two aspects (O'Mahony et al. 2004c). The first one is referred to as accuracy: are the items recommended after the attack actually liked? The second aspect is called stability: does the system suggest different products after the attack? Robustness for a CF system is defined in (O'Mahony et al. 2002a,b), as follows:

Fix a set, $A$, of unrated user-item pairs in the database, which remain unrated in the transformed database i.e. $A \subseteq \left\{(a, j) \,|v_{a,j} = v'_{a,j} = \perp\right\}$. Robustness is defined for the set $A$. Assuming that the set of vote values is bounded above by $R_{max}$ and below by $R_{min}$, for each $(a, j) \in A$, the normalized absolute error, NAE, of prediction pre- and post-attack $T$ is given by

$$NAE\,(a, j, T) = \frac{\left|p_{a,j} - p'_{a,j}\right|}{\max\left\{\left|p_{a,j} - R_{\max}\right|, \left|p_{a,j} - R_{\min}\right|\right\}}$$

The robustness of prediction $\underline{p_{a,j}}$ to attacks of cost $c$ is given by

$$Robust\,(a, j, c) = 1 - \max_{\{T \in \tau : C(T) \leq c\}} NAE\,(a, j, T)$$

Furthermore, the robustness of the CF recommendation system on the set $A$ to attacks of cost $c$ is given by

$$Robust\,(A, c) = \min_{(a,j) \in A} Robust\,(a, j, c)$$

Since different studies focus on different aspects of shilling attacks, there are various measures are used for evaluating the proposed schemes. We group such metrics and listed the most widely used ones in Table 6.

In order to evaluate how effective the shilling attack detection algorithms are, various metrics like precision, recall, F1 metric, and correctness are used, as listed in Table 6. Precision and recall measures how well detection algorithms detect attacks. Precision and recall can be calculated, as follows (Bhaumik et al. 2011; Bryan et al. 2008):

$$Precision = \#true\ positives/(\#true\ positives + \#\ false\ positives)$$
$$Recall = \#true\ positives/(\#true\ positives + \#\ false\ negatives)$$

in which *# true positives* is the number of attack profiles correctly identified as attacks, *# false positives* is the number of authentic profiles that were misclassified, and *# false negatives* is the number of attack profiles that were misclassified. $F1\,measure$ is the combination of *precision* and *recall*, which can be described, as follows:

$$F1 = (2 \times precision \times recall)\ /\ (precision + recall)$$

*Correctness* (He et al. 2010) or *classification accuracy* is the ratio of the number of profiles correctly classified as fake profiles over the number of all profiles. Zhang et al. (2006a,b) propose to use three measures for attack detection: *detection rate*, *false alarm rate*, and ROC. The detection rate or classification accuracy is defined as the number of detected attacks divided by the number of attacks. The false alarm rate is the number of normal profiles that are predicted as attacks divided by the number of normal profiles. The ROC curve measures the extent to which the detection algorithm can successfully distinguish between attacks and normal profiles.

As seen from Table 6, *prediction shift* (Burke et al. 2005b; Burke et al. 2005c), *absolute prediction shift* (Ji et al. 2007), *hit ratio* (Bhaumik et al. 2007a,b), *high rating ratio* (Cheng and Hurley 2009a,b,c), and *ExptopN* (Zhang 2009c; Zhang et al. 2006a) metrics are utilized. Prediction shift is the average change in the predicted rating for the attacked item before and after the attack. Absolute prediction shift measures the distortion of prediction occurring due to an attack (Ji et al. 2007). High rating ratio (Cheng and Hurley 2009a,b,c) shows how much predictions are pushed to high values for an attacked item. Besides providing predictions for single items, CF algorithms also provide a sorted list of $N$ items, which would be liked by a user. Thus, to assess how shilling attacks affect top-$N$ recommendation list, hit ratio and Exptop$N$ measures are used. Hit ratio (Mehta and Nejdl 2008) measures the effect of attack profiles on top-$N$ recommendations. It is the ratio of the number of hits across all users to the number of users. Exptop$N$ (expected top-$N$ occupancy) (Zhang et al. 2006a) is defined as the expected number of occurrences of all target items in a top-$N$ recommendation list.

In addition to the listed ones in Table 6, there are some infrequently used metrics for assessing effects of shilling attacks. *Number of good predictions* (O'Mahony et al. 2006a) is used to calculate the increase in the number of good predictions that are made for targeted items following a push attack. Similarly, nuke attacks can be evaluated with respect to the *number of bad predictions* that are made for targeted items after a nuke attack (O'Mahony et al. 2005). *Mean absolute prediction error* (MAPE) (O'Mahony et al. 2003; O'Mahony 2004; O'Mahony et al. 2004d) measures the prediction shift that has been achieved by an attack. It is defined as the absolute difference between the pre- and post-attack predictions (O'Mahony 2004). NMAPE is used to facilitate a comparison between the robustness of CF systems operating on different rating scales (O'Mahony 2004). It can be computed by dividing the MAPE to the difference between the maximum and the minimum ratings. *Root mean squared distortion* (Van Roy and Yan 2009, 2010; Yan and Van Roy 2009; Yan 2009) is also used to assess the influence due to manipulation. *Mean average error on attacked item* is used to measure the effect of the attack on prediction stability (Mehta et al. 2007b). Lam and Riedl (2004) define *power of attack* (POA), which is the percentage of predictions for target items not manipulated to some target value ($r_{max}$ or $r_{min}$). However, in O'Mahony et al. (2004c), POA is defined as the average change in prediction toward some target value over all target users and items. This metric is similar to prediction shift measure.

To evaluate the robust CF algorithms with respect to accuracy, *mean absolute error* (MAE), *normalized mean absolute error* (NMAE), and so on are utilized. MAE measures how close the estimated predictions to their observed ones. It is used to test how accurate the predictions are. Like NMAPE, NMAE is used to facilitate a comparison between the robustness of CF systems operating on different rating scales. It can be computed by dividing the MAE to the difference between the maximum and the minimum ratings. Besides MAE and NMAE metrics, *root mean squared error* (RMSE) can also be used to measure the accuracy of CF algorithms (Cheng and Hurley 2010b). Another metric that can be used to evaluate a robust algorithm is called *coverage*. *Coverage* (O'Mahony et al. 2006a) is the ratio of the

number of predictions that an algorithm can estimate to the total number of predictions that are requested.

*Mean average error* (Mehta et al. 2007b) can be used for capturing the deviation from actual values. It can be computed, as follows:

$$Mean\,average\,error = 1/M \times |p - p_o|,$$

where $p$ represent predicted rating, $p_o$ is the observed vote, and $M$ shows number of ratings provided by the active user. Mehta et al. (2007b) also propose to utilize *root mean average error* to find out the ability of CF scheme to generalize and highlight large errors.

Detection algorithms and effects of shilling attacks are investigated with respect to the two parameters, *attack size* and *filler size*, of the detection algorithms and attack profiles. *Attack size* can be measured as a percentage of the pre-attack user count (Bhaumik et al. 2006). For example, if there are 100 users' profiles in the user-item matrix, thus, an attack size of 5 % corresponds to five attack profiles added to the database. *Filler size* refers to the number of unrated cells chosen to be filled with fake ratings while creating the attacking profile (Hurley et al. 2009). In some studies (Hurley et al. 2007; O'Mahony et al. 2006b), attack size refers to the filler size, i.e. shows the number of ratings inserted during the course of an attack.

In addition to accuracy analysis, cost-benefit analysis is also conducted to find out the substantial profits, which can be realized by the attackers. O'Mahony et al. (2006b) utilize the ROI metric for cost-benefit analysis. ROI shows the earnings from invested capital expressed as a portion of the outlay; and can be computed, as follows:

$$\text{ROI} = (\textit{total benefits} - \textit{total costs}) \,/\, (\textit{total costs})$$

Hurley et al. (2007) examine shilling attacks with respect to cost. They present a framework for quantifying the benefits attackers get, taking into account the financial cost of mounting the attack. The authors also utilize the concept of ROI for cost-benefit analysis.

## 6 Discussion-open questions

The Internet has been receiving increasing attention since the beginning. It seems that this popularity will continue to grow. Due to the popularity of the Internet, its increasing availability, and easy access to it, traditional commerce has been increasingly replaced by electronic commerce. As long as customers trade over the Internet via online vendors, number of users accessing the Internet and amount of products available online will constantly increase; and still lead information overload problem. Then, to overcome the information overload problem, recommender systems will be widely used by many online vendors. Therefore, the research about recommender systems seems to remain popular. Similarly, shilling attacks against such systems will be in place, as well.

Many studies about shilling attacks take attention of the researchers. Although many things have been achieved, there are still missing gaps that should be filled. We can summarize the open questions about shilling attacks, as follows:

1. Shilling attacks against numeric ratings-based CF schemes have been extensively studied. Users' preferences might be represented using either numeric or binary ratings. However, there is one study (Long and Hu 2010) only discussing how to attack binary ratings-based recommendation systems. Hence, shilling attacks strategies against binary ratings-based recommendation algorithms need to be further investigated.

2. Many researchers have investigated how to attack memory-based CF schemes (correlation-based systems-either user-user or item-item) deeply. And little work has been done about attacking model-based CF systems. In addition to memory-based schemes, model-based and hybrid CF approaches are also widely used. Thus, further researches should focus on developing attack strategies against model-based and hybrid CF algorithms.
3. Recommender systems can be CF schemes based on users ratings or content-based schemes on product contents. Researchers usually have focused on attacking CF schemes only. Content-based systems are rarely scrutinized. Thus, more work should be conducted to develop attack strategies against content-based or hybrid (CF and content-based) approaches.
4. Once the new attack strategies are developed for the abovementioned algorithms (content-based or hybrid schemes), existing detection algorithms should be investigated to find out whether they can detect such new attacks or not. If they are not able to detect them, related shilling attack detection algorithms should be developed. Since known detection algorithms are designed to detect shilling attacks created for attacking collaborative filtering schemes, they might not work for detecting the new strategies designed to attack content-based or hybrid schemes.
5. Extensive evaluation has been performed using some benchmark data sets and evaluation metrics. Considering the real data sets, very sparse data sets have been utilized. However, dense data sets should also be used to assess the shilling attacks, detection algorithms, and robust CF schemes. Therefore, more evaluations are expected to done using dense data sets like Jester.
6. Creating a successful attack strategy mainly depends on the available knowledge about the targetted system. In order to prevent the attackers from developing successful attack strategies, more research should be done to prevent them from obtaining useful information about the targetted system.
7. Providing recommendations while preserving privacy is also receiving attention. Privacy-preserving CF schemes protect users' privacy. However, they might be vulnerable against shilling attacks. Therefore, such schemes should be investigated to figure out whether they are vulnerable against shilling attacks or not.
8. In the literature, there are only couple of studies discussing the cost-benefit analysis of shilling attacks. Detailed cost-benefit analysis should be performed for each possible shilling attack type.

## 7 Conclusions and future work

In this survey, we first briefly discussed the related survey papers and described the missing scopes. Second, we covered possible shilling attack types and explained them briefly. Third, we grouped shilling attacks according to some dimensions and introduced more classification dimensions. Fourth, we discussed the related research about shilling attacks covering the studies focusing on shilling attacks, detection algorithms, robustness analysis, robust algorithms, and cost-benefit analysis. Fifth, we briefly presented some statistical analyses about the studies. Sixth, we discussed evaluation metrics utilized to assess the effects of shilling attacks, detection attacks, and performance of robust algorithms. We finally discussed some missing works that should be completed for further research.

Although we surveyed about shilling attacks against CF schemes in general, we are planning to conduct detailed separate surveys about shilling attacks strategies, detection algorithms, and robustness analysis and robust algorithms, respectively. Since we introduced

various classification dimensions and the related attributes for grouping shilling attacks, we will classify the known attacks according to such dimensions. More work need to be done to create new attack strategies and the related detection algorithms to detect them. Moreover, additional study should be done to enhance the robustness of well-known model-based and hybrid CF algorithms.

**Acknowledgments** This work is partially supported by the Grant 111E218 from TUBITAK.

## References

ACM (1992) Special issue on information filtering. Commun ACM 35(12)

Bhaumik R, Williams CA, Mobasher B, Burke RD (2006) Securing collaborative filtering against malicious attacks through anomaly detection. In: Proceedings of the 4th workshop on intelligent techniques for web personalization, Boston, MA

Bhaumik R, Burke RD, Mobasher B (2007a) Effectiveness of crawling attacks against web-based recommender systems. In: Proceedings of the 5th workshop on intelligent techniques for web personalization, Vancouver, BC, Canada, pp 17–26

Bhaumik R, Burke RD, Mobasher B (2007b) Crawling attacks against web-based recommender systems. In: Proceedings of the international conference on data mining, Las Vegas, NV, USA, pp 183–189

Bhaumik R, Mobasher B, Burke RD (2011) A clustering approach to unsupervised attack detection in collaborative recommender systems. In: Proceedings of the 7th IEEE international conference on data mining, Las Vegas, NV, USA, pp 181–187

Breese JS, Heckerman D, Kadie K (1998) Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the 14th conference on uncertainty in artificial intelligence, Madison, WI, USA, pp 43–52

Bryan K, O'Mahony MP, Cunningham P (2008) Unsupervised retrieval of attack profiles in collaborative recommender systems. In: Proceedings of the 2nd ACM international conference on recommender systems, Lausanne, Switzerland, pp 155–162

Burke RD, Mobasher B, Zabicki R, Bhaumik R (2005a) Identifying attack models for secure recommendation. In: Proceedings of the WebKDD workshop on the next generation of recommender systems research, San Diego, CA, USA, pp 19–25

Burke RD, Mobasher B, Bhaumik R (2005b) Limited knowledge shilling attacks in collaborative filtering systems. In: Proceedings of workshop on intelligent techniques for web personalization, Edinburgh, UK

Burke RD, Mobasher B, Bhaumik R, Williams CA (2005c) Segment-based injection attacks against collaborative filtering recommender systems. In: Proceedings of the 5th IEEE international conference on data mining, Houston, TX, USA, pp 577–580

Burke RD, Mobasher B, Bhaumik R, Williams CA (2005d) Collaborative recommendation vulnerability to focused bias injection attacks. In: Proceedings of the Workshop on privacy and security aspects of data mining, Houston, TX, USA, pp 35–43

Burke RD, Mobasher B, Williams CA, Bhaumik R (2006a) Classification features for attack detection in collaborative recommender systems. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining, Philadelphia, PA, USA, pp 542–547

Burke RD, Mobasher B, Williams CA, Bhaumik R (2006b) Detecting profile injection attacks in collaborative recommender systems. In: Proceedings of the 8th IEEE conference on e-commerce technology, San Francisco, CA, USA, pp 23–30

Burke RD, O'Mahony MP, Hurley NJ (2011) Robust collaborative recommendation. In: Ricci F, Rokach L, Shapira B, Kantor PB (eds) Recommender systems handbook. Springer, New York, pp 805–835

Cheng Z, Hurley NJ (2009a) Robustness analysis of model-based collaborative filtering systems. Lect Notes Comput Sci 6206:3–15

Cheng Z, Hurley NJ (2009b) Effective diverse and obfuscated attacks on model-based recommender systems. In: Proceedings of the 3rd ACM international conference on recommender systems, New York, NY, USA, pp 141–148

Cheng Z, Hurley NJ (2009c) Trading robustness for privacy in decentralized recommender systems. In: Proceedings of the 31st conference on innovative applications of artificial intelligence, Pasadena, CA, USA, pp 79–84

Cheng Z, Hurley NJ (2010a) Analysis of robustness in trust-based recommender systems. In: Proceedings of the 9th conference on adaptivity, personalization and fusion of heterogeneous information, Paris, France, pp 114–121

Cheng Z, Hurley NJ (2010b) Robust collaborative recommendation by least trimmed squares matrix factorization. In: Proceedings of the 22nd IEEE international conference on tools with artificial intelligence, Arras, France, pp 105–112

Chirita PA, Nejdl W, Zamfir C (2005) Preventing shilling attacks in online recommender systems. In: Proceedings of the 7th annual ACM international workshop on web information and data management, Bremen, Germany, pp 67–74

Dellarocas C (2000) Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In: Proceedings of the 2nd ACM conference on electronic commerce, Minneapolis, MN, USA, pp 150–157

Goldberg D, Nichols D, Oki BM (1992) Using collaborative filtering to weave an information tapestry. Commun ACM 35(12):61–70

He F, Wang X, Liu B (2010) Attack detection by rough set theory in recommendation system. In: Proceedings of the IEEE international conference on granular computing, San Jose, CA, USA, pp 692–695

Herlocker JL, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. ACM Trans Inf Syst 22(1):5–53

Hurley NJ, O'Mahony MP, Silvestre GCM (2007) Attacking recommender systems: a cost-benefit analysis. IEEE Intell Syst 22(3):64–68

Hurley NJ, Cheng Z, Zhang M (2009) Statistical attack detection. In: Proceedings of the 3rd ACM international conference on recommender systems, New York, NY, USA, pp 149–156

Ji AT, Yeon C, Kim HN, Jo GS (2007) Distributed collaborative filtering for robust recommendations against shilling attacks. Lect Notes Comput Sci 4509:14–25

Lam SK, Riedl JT (2004) Shilling recommender systems for fun and profit. In: Proceedings of the 13th international conference on world wide web, New York, NY, USA, pp 393–402

Lam SK, Riedl JT (2005) Privacy, shilling, and the value of information in recommender systems. In: Proceedings of the user modeling workshop on privacy-enhanced personalization, Edinburgh, UK, pp 85–92

Lam SK, Frankowski D, Riedl JT (2006) Do you trust your recommendations? An exploration of security and privacy issues in recommender systems. Lect Notes Comput Sci 3995:14–29

Lang J, Spear M, Wu SF (2010) Social manipulation of online recommender systems. Lect Notes Comput Sci 6430:125–139

Li C, Luo Z (2011) Detection of shilling attacks in collaborative filtering recommender systems. In: Proceedings of the international conference of soft computing and pattern recognition, Dalian, China, pp 190–193

Long Q, Hu Q (2010) Robust evaluation of binary collaborative recommendation under profile injection attack. In: Proceedings of the IEEE international conference on progress in informatics and computing, Shanghai, China, pp 1246–1250

Massa P, Avesani P (2007) Trust-aware recommender systems. In: Proceedings of the 1st ACM international conference on recommender systems, Minneapolis, MN, USA, pp 17–24

Mehta B (2007) Unsupervised shilling detection for collaborative filtering. In: Proceedings of the 22nd international conference on artificial intelligence, Vancouver, BC, Canada, pp 1402–1407

Mehta B, Hofmann T (2008) A survey of attack-resistant collaborative filtering algorithms. IEEE Data Eng Bull 31(2):14–22

Mehta B, Nejdl W (2008) Attack resistant collaborative filtering. In: Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval, Singapore, pp 75–82

Mehta B, Nejdl W (2009) Unsupervised strategies for shilling detection and robust collaborative filtering. User Model User Adapt Interact 19(1-2):65–97

Mehta B, Hofmann T, Nejdl W (2007a) Lies and propaganda: Detecting spam users in collaborative filtering. In: Proceedings of the 12th international conference on intelligent user interfaces, Honolulu, HI, USA, pp 14–21

Mehta B, Hofmann T, Nejdl W (2007b) Robust collaborative filtering. In: Proceedings of the 1st ACM international conference on recommender systems, Minneapolis, MN, USA, pp 49–56

Miyahara K, Pazzani MJ (2002) Improvement of collaborative filtering with the simple Bayesian classifier. IPSJ J 43(11)

Mobasher B, Burke RD, Bhaumik R, Williams CA (2005) Effective attack models for shilling item-based collaborative filtering systems. In: Proceedings of the WebKDD workshop, Chicago, IL, USA

Mobasher B, Burke RD, Williams CA, Bhaumik R (2006a) Analysis and detection of segment-focused attacks against collaborative recommendation. Lect Notes Comput Sci 4198:96–118

Mobasher B, Burke RD, Sandvig JJ (2006b) Model-based collaborative filtering as a defense against profile injection attacks. In: Proceedings of the 21st national conference on artificial intelligence, Boston, MA, USA, pp 1388–1393

Mobasher B, Burke RD, Bhaumik R, Sandvig JJ (2007a) Attacks and remedies in collaborative recommendation. IEEE Intell Syst 22(3):56–63

Mobasher B, Burke RD, Bhaumik R, Williams CA (2007b) Towards trustworthy recommender systems: an analysis of attack models and algorithm robustness. ACM Trans Internet Technol 7(4):23–60

O'Donovan J, Smyth B (2006) Is trust robust?: An analysis of trust-based recommendation. In: Proceedings of the 11th international conference on intelligent user interfaces, Sydney, Australia, pp 101–108

O'Mahony MP, Hurley NJ, Silvestre GCM (2002a) Towards robust collaborative filtering. Lect Notes Comput Sci 2464:87–94

O'Mahony MP, Hurley NJ, Silvestre GCM (2002b) Promoting recommendations: an attack on collaborative filtering. In: Proceedings of the 13th international conference on database and expert systems applications, Aix-en-Provence, France, pp 494–503

O'Mahony MP, Hurley NJ, Silvestre GCM (2003) Collaborative filtering-safe and sound. Lect Notes Comput Sci 2871:506–510

O'Mahony MP (2004) Towards robust and efficient automated collaborative filtering. PhD dissertation, University College Dublin

O'Mahony MP, Hurley NJ, Silvestre GCM (2004a) Utility-based neighborhood formation for efficient and robust collaborative filtering. In: Proceedings of the 5th ACM conference on electronic commerce, New York, NY, USA, pp 260–261

O'Mahony MP, Hurley NJ, Silvestre GCM (2004b) Efficient and secure collaborative filtering through intelligent neighbor selection. In: Proceedings of the 16th European conference on artificial intelligence, Valencia, Spain, pp 383–387

O'Mahony MP, Hurley NJ, Kushmerick N, Silvestre GCM (2004c) Collaborative recommendation: a robustness analysis. ACM Trans Internet Technol 4(4):344–377

O'Mahony MP, Hurley NJ, Silvestre GCM (2004d) An evaluation of neighborhood formation on the performance of collaborative filtering. Artif Intell Rev 21(3-4):215–228

O'Mahony MP, Hurley NJ, Silvestre GCM (2005) Recommender systems: Attack types and strategies. In: Proceedings of the 20th national conference on artificial intelligence, Pittsburgh, PA, USA, pp 334–339

O'Mahony MP, Hurley NJ, Silvestre GCM (2006a) Detecting noise in recommender system databases. In: Proceedings of the 11th international conference on intelligent user interfaces, Sydney, Australia, pp 109–115

O'Mahony MP, Hurley NJ, Silvestre GCM (2006b) Attacking recommender systems: The cost of promotion. In: Proceedings of the workshop on recommender systems, in conjunction with the 17th European conference on artificial intelligence, Riva del Garda, Trentino, Italy, pp 24–28

O'Mahony MP, Smyth B (2007a) Evaluating the robustness of collaborative web search. In: Proceedings of the 18th Irish conference on artificial intelligence and cognitive science, Dublin, Ireland

O'Mahony MP, Smyth B (2007b) Collaborative web search: a robustness analysis. Artif Intell Rev 28(1): 69–86

Oostendorp N, Sami R (2009) The copied-item injection attack. In: Proceedings of the workshop on recommender systems and the social web, New York, NY, USA, pp 63–70

Pennock DM, Horvitz E, Lawrence S, Giles CL (2000) Collaborative filtering by personality diagnosis: a hybrid memory- and model-based approach. In: Proceedings of the 16th conference on uncertainty in artificial intelligence, Stanford, CA, USA, pp 473–480

Polat H, Du W (2005) Privacy-preserving collaborative filtering. Int J Electron Commer 9(4):9–35

Ray S, Mahanti A (2009a) Filler item strategies for shilling attacks against recommender systems. In: Proceedings of the 42nd Hawaii international conference on system sciences. Big Island, HI, USA, pp 1–10

Ray S, Mahanti A (2009b) Strategies for effective shilling attacks against recommender systems. Lect Notes Comput Sci 5456:111–125

Ray S, Mahanti A (2010) Improving prediction accuracy in trust-aware recommender systems. In: Proceedings of the 43rd Hawaii international conference on system sciences, Kauai, HI, USA, pp 1–9

Ramezani M, Sandvig JJ, Schimoler T, Gemmell J, Mobasher B, Burke RD (2009) Evaluating the impact of attacks in collaborative tagging environments. In: Proceedings of the international conference on computational science and engineering, Vancouver, BC, Canada, pp 136–143

Resnick P, Sami R (2007) The influence-limiter: Provably manipulation-resistant recommender systems. In: Proceedings of the 1st ACM international conference on recommender systems, Minneapolis, MN, USA, pp 25–32

Resnick P, Sami R (2008a) Manipulation-resistant recommender systems through influence limits. ACM SIGecom Exch 17(3):1–4
Resnick P, Sami R (2008b) The information cost of manipulation resistance in recommender systems. In: Proceedings of the 2nd ACM international conference on recommender systems. Lausanne, Switzerland, pp 147–154
Sandvig JJ, Mobasher B, Burke RD (2007a) Robustness of collaborative recommendation based on association rule mining. In: Proceedings of the 1st ACM conference on recommender systems, Minneapolis, MN, USA, pp 105–112
Sandvig JJ, Mobasher B, Burke RD (2007b) Impact of relevance measures on the robustness and accuracy of collaborative filtering. In: Proceedings of the 8th international conference on electronic commerce and web technologies, Regensburg, Germany, pp 99–108
Sandvig JJ, Mobasher B, Burke RD (2008) A survey of collaborative recommendation and the robustness of model-based algorithms. IEEE Data Engineering Bulletin 31(2):3–13
Su XF, Zeng HJ, Chen Z (2005) Finding group shilling in recommendation system. In: Proceedings of the 14th international conference on world wide web, Chiba, Japan, pp 960–961
Tang T, Tang Y (2011) An effective recommender attack detection method based on time SFM factors. In: Proceedings of the IEEE 3rd international conference on communication software and networks, Xi'an, China, pp 78–81
Van Roy B, Yan X (2009) Manipulation-resistant collaborative filtering systems. In: Proceedings of the 3rd ACM conference on recommender systems, New York, NY, USA, pp 165–172
Van Roy B, Yan X (2010) Manipulation robustness of collaborative filtering. Manag Sci 56(11):1911–1929
Williams CA (2006) Profile injection attack detection for securing collaborative recommender systems. Masters thesis, DePaul University
Williams CA, Bhaumik R, Burke RD, Mobasher B (2006a) The impact of attack profile classification on the robustness of collaborative recommendation. In: Proceedings of the WebKDD workshop, Philadelphia, PA, USA
Williams CA, Mobasher B, Burke RD, Bhaumik R, Sandvig JJ (2006b) Detection of obfuscated attacks in collaborative recommender systems. In: Proceedings of the workshop on recommender systems, in conjunction with the 17th European conference on artificial intelligence, Riva del Garda, Trentino, Italy, pp 19–23
Williams CA, Mobasher B, Burke RD (2007a) Defending recommender systems: detection of profile injection attacks. Serv Oriented Comput Appl 1(3):157–170
Williams CA, Mobasher B, Burke RD, Bhaumik R (2007b) Detecting profile injection attacks in collaborative filtering: a classification-based approach. Lect Notes Comput Sci 4811:167–186
Wu Z, Cao J, Mao B, Wang Y (2011) Semi-SAD: applying semi-supervised learning to shilling attack detection. In: Proceedings of the 5th ACM conference on recommender systems, Chicago, IL, USA, pp 289–292
Yan X (2009) Manipulation robustness of collaborative filtering systems. PhD dissertation, Stanford University
Yan X, Van Roy B (2009) Manipulation robustness of collaborative filtering systems. The Computing Research Repository (abs/0903.0069)
Zhang FG (2008) Analysis of segment shilling attack against trust based recommender systems. In: Proceedings of the 4th international conference on wireless communications, networking and mobile computing, Dalian, China, pp 1–4
Zhang FG (2009a) Reverse bandwagon profile injection attack against recommender systems. In: Proceedings of the 2nd international symposium on computational intelligence and design, Changsha, China, pp 15–18
Zhang FG (2009b) Average shilling attack against trust-based recommender systems. In: Proceedings of the international conference on information management, innovation management and industrial engineering, Xi'an, China, pp 588–591
Zhang FG (2009c) A survey of shilling attacks in collaborative filtering recommender systems. In: Proceedings of the international conference on computational intelligence and software engineering, Wuhan, China, pp 1–4
Zhang FG (2010) Analysis of love-hate shilling attack against e-commerce recommender system. In: Proceedings of the international conference of information science and management engineering, Xi'an, China, pp 318–321
Zhang FG (2011a) Preventing recommendation attack in trust-based recommender systems. J Comput Sci Technol 26(5):823–828
Zhang FG (2011b) Analysis of bandwagon and average hybrid attack model against trust-based recommender systems. In: Proceedings of the 5th international conference on management of e-commerce and e-government, Hubei, China, pp 269–273

Zhang FG, Xu SH (2007) Analysis of trust-based e-commerce recommender systems under recommendation attacks. In: Proceedings of the 1st international symposium on data, privacy, and e-commerce, Chengdu, China, pp 385–390

Zhang S, Ouyang Y, Ford J, Makedon F (2006a) Analysis of a low-dimensional linear model under recommendation attacks. In: Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval, Seattle, WA, USA, pp 517–524

Zhang S, Chakrabarti A, Ford J, Makedon F (2006b) Attack detection in time series for recommender systems. In: Proceedings the 20th ACM SIGKDD international conference on knowledge discovery and data mining, Philadelphia, PA, USA, pp 809–814

Zhang Q, Luo Y, Weng C, Li M (2009) A trust-based detecting mechanism against profile injection attacks in recommender systems. In: Proceedings of the 3rd IEEE international conference on secure software integration and reliability improvement, Shanghai, China, pp 59–64