Decision Support Systems 124 (2019) 113112

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

Trustworthy and profit: A new value-based neighbor selection method in recommender systems under shilling attacks

Yuanfeng Cai[a], Dan Zhu[b,*]

[a] Zicklin School of Business, City University of New York–Baruch College, New York, NY 10010, United States of America
[b] Ivy College of Business, Iowa State University, Ames, IA 50010, United States of America

ARTICLE INFO

Keywords:
Recommender systems
Shilling attacks
Profitability
Collaborative filtering

ABSTRACT

User-based collaborative filtering recommender systems are widely deployed by e-retailers to facilitate customer' decision-making and enhance e-retailers' profitability. Despite the advantages these systems provide, their recommendation effectiveness is vulnerable to attacks from malicious users who inject biased ratings. Such attacks against recommender systems are called shilling attacks. Although several shilling attack detection mechanisms have been proposed in previous studies, their detection performance is limited in various attack conditions. Furthermore, few of these mechanisms consider the value-dimension associated with recommendations, which is crucial for e-retailers. This research proposes a novel approach called Value-based Neighbor Selection (VNS) to address the above limitations. The objective of this approach is to protect recommender systems from shilling attacks while improving e-retailers' profitability. It alleviates the aforementioned problems through strategically selecting neighbors whose preferences are then used to make recommendations. We have performed a series of empirical validations in various attack conditions to compare the performance of the proposed method and three benchmark methods, in terms of both recommendation accuracy and e-retailer profitability. The results show the advantages of the proposed method in balancing customer satisfaction and e-retailer profitability.

1. Introduction

The rapid growth of the Internet has enabled access to a vast amount of product information. Although the volume of information creates business opportunities for e-retailers, it also results in information overload, which becomes a burden for customers locating their needs. To deal with this growing problem, e-retailers, such as Amazon and Netflix, are adopting recommender systems to reduce search complexity, facilitate customers' decision making and drive sales [1–3]. Recommender systems use a variety of algorithms, among which *user-based collaborative filtering* is one of the most popular [4–6]. According to this algorithm, users who have shared similar tastes in the past tend to share similar tastes in the future [4,7]. Accordingly, collaborative filtering recommender systems predict customers' purchasing preferences based on analysis of opinions of other customers with similar tastes [1,2,8]. We term a user whose preference is predicted by the system an *active user* [4,7]. For active users, collaborative filtering recommender systems select a list of users with similar preferences. The selected users are called *neighbors*, and preference predictions for active users are based on the explicit preference ratings of their neighbors. Therefore, the quality of recommendations relies on the selection of neighbors used to make the predictions.

Because traditional collaborative filtering recommender systems have been devoted to providing the most accurate recommendations, they have selected neighbors according to similarities in historical rating patterns. Although this approach considers the relevance of recommended items and user satisfaction, its quality is vulnerable to dishonest ratings which prompt misleading recommendations. Dishonest ratings in recommender systems come from two sources: malicious or non-malicious users [6]. Non-malicious users provide erroneous ratings involuntarily, which are called natural noise [6]. Malicious users inject dishonest ratings to deliberately seek improper benefits through recommendations [9]; they either promote their own products to active users, or demote competitors' products [10]. These users are known as *shilling attackers*, and their activities are known as *shilling attacks* [9–12]. Though natural noise also has influences on recommendation results, this study focuses on shilling attacks since it is a more serious concern for recommendation accuracy [13]. As will be discussed in the next section, shilling attackers usually attempt to maintain a relatively high rating similarity in their non-target items with other users in the system, so the system is likely to select them as neighbors and create inappropriate recommendations. Therefore, with

* Corresponding author.
E-mail addresses: Yuanfeng.Cai@baruch.cuny.edu (Y. Cai), dzhu@iastate.edu (D. Zhu).

https://doi.org/10.1016/j.dss.2019.113112
Received 1 March 2019; Received in revised form 14 July 2019; Accepted 14 July 2019
Available online 19 July 2019
0167-9236/ © 2019 Elsevier B.V. All rights reserved.

regard to neighbor selection, it is necessary to consider the likelihood that a user is an attacker.

Although several studies have designed strategies to cope with shilling attacks [5,11,14–21], they suffer from two major limitations. First, they often feature high false-positive rates for attacker detection, and they make binary decisions about users, causing those identified as attackers to be removed from the system. Recommendation accuracy then deteriorates, due to the permanent loss of misidentified genuine users. Second, they do not consider the profits associated with recommendations; they focus only on prediction accuracy, which is important but not sufficient for e-retailers. In reality, an e-retailer's ultimate goal is to enhance their profitability [22]. Recommendations in real-world recommender systems need to balance accuracy and profitability [23]. Although some other scholars have noted the importance of profitability [24,25], their approaches rely on neighbors' preferences to generate profitable recommendations; they do not consider the influence of shilling attacks and assume that each user is a genuine user. Once a neighbor's preference is misled by attackers, recommendation quality is negatively affected.

In this study, we propose an improved approach to address the above limitations that integrates profit into the consideration of user-based collaborative filtering recommender systems under shilling attacks. For each active user, we derive quality recommendations by selecting appropriate neighbors to control the influence of attackers while maximizing the e-retailer's expected profits. In the proposed algorithm, we begin by examining each user's potential as an attacker. To avoid the loss of misidentified genuine users, we estimate the attack probability of each user instead of making a binary filtering decision, as previous studies have done [5,18,21]. For each active user, we find an optimal set of neighbors whose preferences we use to make profitable recommendations, while keeping the potential influence from attackers in this set under control. Finally, we rely on the selected neighbors to predict the preference for each active user for items to be recommended. We conduct experimental studies to empirically test the effectiveness of our proposed approach in various attack scenarios. We compare our approach with various benchmark methods, using both accuracy and profit criteria. The experimental results indicate that our approach increases e-retailers' expected profits without decreasing recommendation accuracy; it provides insights into the design of profitable, trustworthy recommender systems. To the best of our knowledge, this research is the first to strike a balance between customer satisfaction and e-retailer profitability in the case of shilling attacks.

The rest of the paper is organized as follows. We review related literature in Section 2. Section 3 introduces our proposed approach and provides detailed analysis. After we describe our evaluation design and discuss the experimental results using accuracy-based benchmark methods in Section 4, we present a performance evaluation using the profit-based benchmark method in Section 5. In Section 6, we further evaluate the advantages of the proposed method in a more complicated attack scenario. We conclude the paper in Section 7 with a discussion of our study's implications and limitations.

2. Literature review

Recommender systems are important decision support tools which provide personalized service to customers. They explore customers' purchasing interests and help them discover the most suitable products efficiently [26]. Despite the growing popularity of these recommender systems, designing an effective one is still an important research area in information science and decision support systems [25,27]. E-retailers deploy recommender systems to improve customer satisfaction and drive profitability [6]. However, to realize their goals, traditional collaborative filtering recommender systems face two challenges: *the ignorance of the value-dimension of recommendations and shilling attacks.* The first issue affects e-retailer profitability and the second one deteriorates recommendation accuracy, which in turn, damages the trustworthiness of systems. In this section, we review previous research on the two issues and explain the theoretical foundation of the collaborative filtering recommendation algorithm. Finally, we discuss the research gap and the potential contributions of the present study.

2.1. *Value-based recommendation*

The majority of recommender systems are designed to solely encourage recommendation accuracy. Recommendation accuracy is very important to customers' purchasing probability but may not be sufficient to e-retailers' profitability [28]—most accurate recommendation may not be the most profitable one. The case of Netflix [23] clearly confirms this point and indicates the necessity of considering sustainable profitability when designing recommender systems. The Netflix recommender system aims to understand users' preferences accurately in order to recommend the most suitable film. However, films recommended to users may not be the exact same as the direct output of the recommender system. The system avoids the recommendation of high-demand new releases because of the higher costs of acquiring such movies. Recommending these higher-profile films may increase recommendation accuracy, but would hurt profitability. The system maintains recommendation accuracy in lower-profile films to improve customer satisfaction. Hereby, from the e-retailer's perspective, a "good" recommendation should balance both recommendation accuracy and e-retailer profitability.

Several scholars have noted the direct profitability associated with recommendations. Shani et al. [29] propose a Markov decision process model that considers the expected profit from each recommendation. Liu and Shih [30] suggest that recommendations can be made according to a customer's lifetime value—that is, according to recency, frequency, and monetary values. Akoglu and Faloutsos [31] propose that a profitable recommendation can be made if social network relationship information among users is available. Jiang et al. [28] develop a dynamic pricing strategy to recommend products that maximizes e-retailer profits when recommender systems provide the opportunity to interact with customers. Ge et al. [32] design several cost-aware models for travel tour recommendations. Even though the aforementioned studies take profitability into account, users may lose confidence in such recommender systems if they totally ignore user preferences and recommendation accuracy. Some other scholars have proposed recommendation models that consider both user preferences and profitability, theoretically [22,33] and empirically [24,25,34,35]. For example, Chen et al. [24] develop two types of recommender systems: a *hybrid perspective recommender system* (HPRS) and a *convenience plus profitability perspective recommender system* (CPPRS), both of which recommend items with the largest (purchasing probability × item profit). Other studies propose methods which improve both recommendation accuracy and aggregate diversity [34,35]. An increase in aggregate diversity has the potential for profitability enhancement through selling more diverse products [35]. Panniello et al. [25] design a system that makes either the most profitable recommendation or the most accurate recommendation, depending on the user's current level of trust in the system. This strategy is appropriate when making longitudinal or long-term recommendations to users, because its recommendations rely on trust levels that change over time.

2.2. *Collaborative filtering recommendation algorithm*

Besides profitability, a good recommendation needs to preserve accuracy, as discussed in Subsection 2.1. Thus, a recommendation algorithm is expected to predict users' actual preference accurately. The two most prominent algorithms to estimate user preferences are content-based and collaborative filtering [2,6]. Content-based filtering utilizes the content features of items that an active user rated before to recommend similar items [27,36]. Collaborative filtering leverages the opinions of other users with similar tastes to make recommendations

[37]. Since collaborative filtering does not require content analysis, it shows advantages in many situations where the content features are hard to extract (e.g., audio and video) [38,39]. Collaborative filtering has been effectively applied to make recommendations in several application domains [37], such as e-commerce [40], e-learning [41], quality of service ranking [42], and others. In this study, we focus on collaborative filtering because of its wider applications.

Collaborative filtering basically imitates human "word-of-mouth" behavior: Customers share opinions on items with others and others determine whether to pursue when making choices [12,39]. Collaborative filtering recommender systems automate this process by collecting users' opinions on items (e.g., rating values) to provide recommendations [38,40]. Collaborative filtering assumes that rating values could reflect users' preferences [43]. The purchasing decision of an active user is likely to be affected by the opinions of "neighbors" with similar tastes [37,43]. There are two types of collaborative filtering algorithms: user-based and item-based [26]. In user-based collaborative filtering, "neighbors" are users who share similar rating preferences to the active user; In item-based collaborative filtering, "neighbors" are items similar to those rated by the active user [38]. Both types of collaborative filtering recommender systems first identify neighbors (e.g., users or items), and aggregate their opinions to predict the preference of an active user on an unrated item [39]. To an active user, items recommended from user-based collaborative filtering are usually those preferred by other similar users, and those from user-based collaborative filtering are similar to what the user liked before [37].

Despite its popularity and widely applications, collaborative filtering suffers from limitations. Since it relies on users' preferences (rating values) to make recommendations, the quality of ratings affect recommendation accuracy [26]. If biased ratings are intentionally injected, recommendation accuracy will be severely undermined. Thus, collaborative filtering is vulnerable to shilling attacks [5]. In recommender systems, there is a natural incentive for owners of products to have their own products be recommended more frequently than those of competitors [18]. Intuitively, they could improve the quality of their own products so that the products could be preferred by more users and thus more likely to be recommended [9]. However, someone may choose a deceitful but quick way, e.g., launching an attack, to influence recommendation frequency and seek improper benefits [10]. Because of the open nature of collaborative filtering recommender systems, it is relatively easy to inject attackers' rating profiles [44]. Different from genuine users' profiles, which usually reflect their real preferences, attackers' profiles consist of unfair ratings aiming to mislead genuine users [19]. Such profiles injection attacks are called shilling attacks [12,21]. Shilling attacks are common occurrences in recommender systems and several cases have been reported in prior literature [9,21]. For examples, Sony Pictures insert counterfeit opinions into recommender systems to promote their new released films [5]; Top users in Digg.com conspire to promote their own articles by providing unfair votes to the system [45]; Motivated to increase sales, authors in Amazon.com introduce biased ratings into the system to artificially promote their own books [44]. Genuine users will lose faith in the system if shilling attackers continuously achieve their goals, so it is crucial to resist them.

Previous studies have found that user-based collaborative filtering is more susceptible to shilling attacks than item-based collaborative filtering [9,12]. Because of its higher susceptibility, when addressing the issues of shilling attacks and value-based recommendations in this study, we employ a user-based collaborative filtering algorithm to make recommendations. Specifically, for two users, u_t and u_k, we use Pearson correlation coefficient (PCC) [7,8] to calculate their taste similarity as:

$$w(t,k) = \frac{\sum_{j\in J}(r_{t,j} - \bar{r}_t)(r_{k,j} - \bar{r}_k)}{\sqrt{\sum_{j\in J}(r_{t,j} - \bar{r}_t)^2 \sum_j (r_{k,j} - \bar{r}_k)^2}} \quad (1)$$

where $r_{k,j}$ is the rating of a user u_k on item i_j; $\bar{r}_k$ is the average rating of items for which the user u_k has rated; and J is the set of items rated by both user u_t and u_k. Among other similarity measures, we opt for Pearson correlation coefficient because of its outstanding performance and wide usage in user-based collaborative filtering recommender systems, which makes the proposed method more easily be applied [7,36,46–48]. An active user u_t's predicted rating $r_{t,a}$ of a new item a is computed on the basis of the user's neighbors' preferences and the user's rating similarity with neighbors, as defined in Eq. (2):

$$r_{t,a} = \bar{r}_t + \kappa_t \sum_{k\in K} w(t,k)(r_{k,a} - \bar{r}_k) \quad (2)$$

where $k \in K$ is a set of K neighbors; $\kappa_t = 1/\Sigma_{a=1}{}^{m}|w(t,k)|$ is a normalizing factor; $\bar{r}_k$, $r_{k,a}$ and $w(t,k)$ are defined as in Eq. (1). Accordingly, in a user-based collaborative filtering recommender system, the preference of an active user for an item is predicted as a weighted average of the user's neighbors' preferences for the item.

2.3. Shilling attack models

To design an effective method to limit the impact of shilling attackers, it is necessary to understand their behavior. Shilling attackers have monetary motivations to deliberately distort the predicted ratings of their target items for their own benefit [9,21]. In terms of intentions, there are two types of attacks: *push attacks* and *nuke attacks* [10]. In push attacks, attackers promote their own items; they give items the maximum rating to make them more likely to be recommended to genuine users. In nuke attacks, they bespatter items of competitors by giving them the minimum rating to make them less likely to be recommended.

If attackers aim to make an item more or less likely to be recommended—that is, elevate or lower its predicted rating to an active user—simply injecting unfair ratings is insufficient. As shown in Subsection 2.2, predicted ratings are determined not only by attackers' preferences but also by similarities between attackers and active users. Thus, to achieve their goal, attackers must also rate non-target items strategically to increase their similarities with genuine active users. Fig. 1 shows a general form of an attack rating profile. The first row lists each item i in the entire set I, and the second row displays the corresponding rating. Table 1 summarizes the notations in Fig. 1. Attackers always rate the target item(s) I_T with either the maximum rating (r_{max}) in a push attack or the minimum rating (r_{min}) in a nuke attack. Moreover, they always have a set of unrated items I_N that they do not rate, since it is unlikely that a user will rate every item in the system. Attackers use different strategies to make selections and give ratings to non-target items, that is, I_S and I_F. Mobasher et al. [12] summarize various types of shilling attack models. We discuss the general

I_S ratings for *s selected* items			I_F ratings for *f filler* items			I_N *n unrated* items			I_T ratings for *t target* items		
i_1^S	…	i_s^S	i_1^F	…	i_f^F	i_1^N	…	i_n^N	i_1^T	…	i_t^T
$\delta(i_1^S)$	…	$\delta(i_s^S)$	$\sigma(i_1^F)$	…	$\sigma(i_f^F)$	null	null	null	$r(i_1)$	...	$r(i_t)$

Fig. 1. The General structure a shilling attack profile.
Adapted from Ref. [12].

Table 1
Explanations of notations in Fig. 1.

Notation	Explanation	Notation	Explanation
I_S	set of *selected* items, $\|I_S\| = s$	$\sigma(\cdot)$	rating for an item in I_F
I_F	set of *filler* items, $\|I_F\| = f$	r_{min}/r_{max}	minimum/ maximum rating in the system
I_N	set of *unrated* items, $\|I_N\| = n$	$r(i_t)$	rating for an item in I_T, $r(i_t) = \begin{cases} r_{min}, & \text{for nuke attack} \\ r_{max}, & \text{for push attack} \end{cases}$
I_S	set of *target* items, $\|I_T\| = t$		
$\delta(\cdot)$	rating for an item in I_S		

characteristics of each model next.

- *Random Attack*: Random attackers enhance their influence on genuine users by filling ratings to a set of randomly chosen items. This set is denoted I_F, and the chosen items are known as *filler items*. To increase similarity with genuine users, the ratings of filler items are expected to be close to those of genuine users. In random attacks, $\sigma(i_j^F)$, $j = 1, \cdots, l$ are generated using a normal distribution $\mathcal{N}(\bar{r}_I, s_I^2)$, where $\bar{r}_I$ is the global mean rating across all items from I_F in the system, and s_I is the global standard deviation of all items from I_F in the system. This type of filling method constitutes the *random filling method* (RFM). Random attackers use only filler items to advance their influence. Thus, $I_S = \emptyset$.
- *Average Attack*: Average attackers have the same profile structures as random attackers. The only difference is the filling method. In average attacks, to generate ratings that are closer to actual ratings, the rating of each filler item $\sigma(i_j^F)$, $j = 1, \cdots, l$ is derived from its individual normal distribution $\mathcal{N}(\bar{r}_i, s_i^2)$, where $\bar{r}_i$ is the average rating of the item i_j^F in the system, and s_i is its standard deviation in the system. This type of filling method is known as the *average filling method* (AFM). Compared with RFM attackers, AFM attackers are more likely to be similar to most genuine users, making themselves more effective.
- *Bandwagon Attack*: Bandwagon attackers strengthen their influence by rating items in addition to I_F. They give r_{max}/r_{min} in push/nuke attacks to a set of *selected items,* denoted as I_S. Items in I_S are not randomly selected but are the most popular items in the system. Accordingly, to increase their similarity, they have more items rated in common with the most genuine users. They use either the AFM or the RFM to fill ratings in I_F. That is, they carry out either AFM bandwagon attacks or RFM bandwagon attacks.
- *Segment Attack*: A segment attack is designed specifically for item-based algorithms [4]. Its profile structure is similar to that of a bandwagon attack, except for the selection of I_S. Unlike attackers in other models, segment attackers target one user group with known preferences. For example, if the target item is a fairy tale, segment attackers aim to influence only a group of users known to be interested in this type of product. Therefore, the items selected are popular among this particular user segment, rather than across the entire system. Items in I_S are rated with r_{max}/r_{min} in push/nuke attacks. Although segment attackers are likely to have a heavy effect on users within a particular segment, they minimize their influence on users outside their target group by rating filler items with r_{min}. Because of their extremely unusual rating patterns—that is, the extreme ratings in filler and selected items—they are usually easily detected [16,18].

2.4. Related research on shilling attack detection

Methods for attack detection have been widely studied. Scholars have analyzed characteristics of various attack types, including phishing [49] and cybersecurity attacks [50], in order to address them. They have proposed detection methods that require a set of cues that correspond to each specific attack type. Thus, the methods cannot be adapted to a different attack type (e.g., shilling attacks) that lack the corresponding cues.

Another line of research has been devoted to design mechanisms to counter natural noise [6,13,26,51,52]. While both natural noise and shilling attack provide untruthful ratings to recommender systems, the two types of untruthful ratings come from different sources with distinct characteristics (i.e., non-malicious and malicious users). Thus, natural noise detection methods cannot be applied directly to shilling attack detection [51,52].

Prior literature also explores detection methods for shilling attacks. One stream of research employs supervised techniques. For example, Chirita et al. [11] create metrics to measure differences between attackers' rating profiles and genuine users' profiles, which can then be used to remove attackers. Williams et al. [16] use C4.5 to develop a method known as the classification-based approach to detect attackers. Zhang and Zhou [19] propose a support vector machine classifier to detect shilling attacks using Hilbert spectrum-based features in rating profiles. Because these supervised detection methods must train classifiers, they are not suitable for detecting unknown types attacks. Some other studies detect attackers using statistical features, such as rating time series discord [14,20]. Such methods are effective when attacks occur within a short period.

Another stream of research on shilling attacks proposes the detection method based on unsupervised techniques. Mehta et al. [15] detect a set of attackers using principal component analysis with the assumption of a known attack size. Bhaumik et al. [17] adopt a k-means clustering approach to find attackers, but it suffers from poor precision. Chung et al. [5] propose a Beta-Protection algorithm based on Beta distribution to remove malicious users, but its detection accuracy is biased in a large attack size. Lee and Zhu [18] analyze the difference between attackers and genuine users as a group and use group-level rating deviation to discriminate attackers. Although their study is one of the most accurate, it suffers from a high false-positive rate when filler or attack sizes are small. A more recent study uses a graph mining method [21] to identify attackers. It assumes that the number of ratings of target items is much larger than that of non-target items, which is valid only when the attack size is quite large.

2.5. Research gap and contribution

An assessment of the literature reveals the limitations of previous research. Although previous studies design strategies for value-based recommendation generation and shilling attack detection respectively, the two issues have not been addressed together. Furthermore, despite the acknowledgement of user preferences, value-based recommender systems designed in the aforementioned studies do not consider vulnerability to shilling attacks [34]. Thus, their recommendation quality is affected by attackers' activities, leaving a research gap. Meanwhile, even the most accurate shilling attack detection mechanisms suffer from high misidentification rates in smaller filler sizes or attack sizes. Removing the ratings of misidentified genuine users further diminishes recommender system performance. Our study aims to address these limitations and fill the research gap. This study contributes to the literature of recommender systems by redesigning the system with the consideration of both e-retailer profitability and the influence of shilling attacks. Moreover, compared to previous shilling attack defense

mechanisms, our proposed method can better protect recommendation accuracy using a probability-based approach instead of a filtering-based one.

3. Value-based neighbor selection method (VNS)

We propose a new method called *value-based neighbor selection* (VNS). This method selects neighbors for active users in user-based collaborative filtering recommender systems under shilling attack. The goal of VNS is to select neighbors who can help increase expected profitability while maintaining recommendation prediction accuracy. To maintain prediction accuracy under shilling attack, the method should be able to identify attackers and mitigate their influence. Thus, the method needs to consider expected profits of recommended items, identification of attackers, and the preference likelihood of recommended items. To avoid losing information from possible misidentified genuine users, we assign each user an estimated probability of being an attacker, rather than making a binary decision about whether to remove the user from the system. In this section, we describe the proposed approach in detail. Although all discussions of methodology and experiments in this study relate to push attacks, our proposed method could be adapted to nuke attacks, which have similar profiles to push attacks but only reverse values in target items.

When selecting neighbors for each active user, VNS accounts for both the active user's preferences and the expected profit from recommendations. Specifically, it follows two principles:

1) To predict an active user's preference more accurately, the influence of attackers should be limited.
2) The expected profit from recommendations generated for the active user, using the selected neighbors, should be maximized.

We formulate the problem as follows: Find a neighbor list for the active user that can maximize expected profit while controlling the level of attackers in the list below threshold τ. The shilling attacker level is defined as the expected number of attackers in the neighbor list. Recognize that in recommender systems, given that users are independent from each other and each user has its own probability of being an attacker, the attacker distribution in the entire user population follows a Poisson binomial distribution, and the shilling attacker level can be evaluated as the sum of each user's probability of being an attacker. We next discuss how to estimate the attacker probability and the procedure of the proposed method.

3.1. Shilling attacker probability

To estimate each user's probability of being an attacker, we use two characteristics of attackers. First, their rating patterns are usually different from those of genuine users; attackers provide unrealistic ratings to their target items, which are usually unpopular items and not preferred by other users. Accordingly, *rating deviation from mean agreement* (RDMA) is developed to detect attackers by examining their rating profile's deviation of agreement from other users [11]. This metric could find abnormal ratings of target items and detect attackers according to the difference between a user's rating and the average rating of items. Although RDMA captures extraordinary ratings in the pattern of an attacker, it is less effective in the case of many attackers [11]. Unfortunately, attackers usually do not appear alone. A group of attackers is more effective in pushing the target item [53]. As shown in Subsection 2.3, a group of attacker profiles are often created with the same target and filler items. Thus, users can be clustered according to their rating patterns. We adopt a modified metric, *group rating deviation from mean agreement* (GRDMA), measured as in Eq. (3), to select groups of users with unusual rating patterns [18]. The rationale of this metric is that a target item(s) is/are usually unpopular, that is, not highly rated by many genuine users [11]. Accordingly, a cluster with many attackers has a relatively higher GRDMA, because it is where users who rated the target item(s) with unusual ratings are gathered. Compared with RDMA, GRDMA magnifies rating deviation in items by emphasizing portions of ratings from attackers on the target item(s). In this way, the cluster(s) with the highest GRDMA would be selected. Thus,

$$GRDMA_C = \max\left\{ |\bar{r}_{C,j} - \bar{r}_j| \bullet \left(\frac{NR_j^C}{NR_j} \right)^2 \right\} j = 1, \cdots, N_C \tag{3}$$

where N_C is the number of items rated by users in the cluster C; $\bar{r}_{C,j}$ is the average rating for item j rated by users in the cluster C and $\bar{r}_j$ is that rated by all users; $NR_j{}^C$ is the number of ratings of item j rated by users in the cluster C; and NR_j is that rated by all users.

Second, though the GRDMA metric can select a group of users with unusual rating patterns, it is less helpful in differentiating attackers from genuine users with special tastes, such as genuine users with similar tastes to attackers. Thus, we consider another feature to further assess which users in the selected group are more likely to be attackers. In recommender systems, the goal of push attackers is not to boost the rating of their target item(s), but to make the item(s) more likely to be recommended to genuine users. To realize this goal, in addition to rate the target items, attackers must give filler items ratings similar to those given by the majority of genuine users. By increasing their rating similarity with genuine users, attackers become more influential to genuine users, such that their target item(s) are more likely to be recommended. Thus, attackers are expected to behave similarly to other users in the system so as to affect recommendations effectively [11,12]; the greater their similarity, the shorter the distance between attackers and genuine users. Therefore, we assess attackers according to their own average distance to other users, using *multidimensional scaling* (MDS) approach [54], a powerful technique that has been applied in previous research [18,55,56]. A user with a smaller average distance is more likely to be an attacker. Thus, we adopt GRDMA along with the average distance to estimate the probability of each user being an attacker. We denote the attacker probability of a user u_k as s_k, where $s_k \in (0, 1)$. The three-step procedure for estimating s_k is as follows:

a. Perform hierarchical clustering for users on the basis of their user–item rating matrix. Calculate $GRDMA_C$ for each cluster. Select the cluster(s) C_{max} that has/have the highest GRDMAs. For users not in the selected cluster(s) C_{max}, set the shilling attack probability to 0 ($s_k = 0$, k = 1, … m, & $k \notin C_{max}$).
b. For users in the selected cluster(s) C_{max}, calculate their user–user rating dissimilarity matrix using (1 − Eq. (1)), such that (1 − w (a, i)). Then perform a classical multidimensional scaling to return the coordinates of every user in the new p-dimensional space. The Euclidean distance between every two users is calculated using the new coordinates. For each user u_k, calculate its average distance from all other users, denoted as $dist_k$ (k = 1, …, m & $k \in C_{max}$).
c. For users in the selected cluster(s) C_{max}, denote the mean and the minimum of average distance values as $dist_{mean}$ and $dist_{min}$. For a user u_k, normalize the $dist_k$ into an attack probability s_k in value [0, 1]. We choose the transformation approach in Eq. (4) such that the user with the lowest average distance has the highest probability of being an attacker. Set the user with a very high average distance (i.e., greater than the mean) to 0 for the probability of being an attacker.

$$\text{For } [dist_{min}, dist_{mean}] \rightarrow [0,1] s_k \quad = 0 \quad \text{if } (dist_k > dist_{mean})$$
$$; s_k = \frac{1}{(e-1)} \left(\frac{dist_{mean} - dist_k}{dist_{mean} - dist_{min} - 1} \right) \quad \text{otherwise} \tag{4}$$

3.2. Value-based neighbor selection (VNS) method

Suppose that the total number of items in the recommender systems is n, and i_j stands for item j, where j = 1, …, n. For an active user u_t, the items to be recommended are denoted as i_1~i_n. Its similarity-based expected profit (SEP) obtained from recommendations is defined as $SEP_t = \sum_{j=l}^{j=n} p_j * l_{t,j}$ [24], where $l_{t,j}$ is the similarity-based likelihood that u_t would prefer item i_j in purchasing, and p_j is the item profit. As shown in Eq. (2), the similarity-based preference likelihood of $u_t$is correlated with both its neighbors' preferences and its rating similarity with neighbors. Thus, it is defined as the Eq. (5) [4,7,24]:

$$l_{t,j} = \kappa_t \sum_{k=1}^{k=m} w(t,k)(r_{k,j} - \bar{r}_k)x_k^t = \frac{\sum_{k=1}^{k=m}(r_{k,j} - \bar{r}_k)*w(t,k)*x_k^t}{\sum_{k=1}^{k=m}|w(t,k)|*x_k^t} \quad (5)$$

where κ_t is the same normalizing factor as in Eq. (1); $w(t,k)$ is the rating similarity between u_t and u_k; $r_{k,j}$ is the rating of a user u_k (k = 1, …, m) on item i_j; $\bar{r}_k$ is the average rating of items on which the user u_k has rated; and $x_k{}^t$ is a binary variable that indicates whether or not a user is selected as a neighbor. Accordingly, the SEP for u_t from the selected neighbors is equivalent to

$$SEP_t = \frac{\sum_{j=l}^{j=n} p_j * \sum_{k=1}^{k=m}(r_{k,j} - \bar{r}_k)*w(t,k)*x_k^t}{\sum_{k=1}^{k=m}|w(t,k)|*x_k^t} \quad (6)$$

where p_j is the profit of item i_j, $j \in [l,n]$, $r_{k,j}$, $\bar{r}_k$, $w(t,k)$ and $x_k{}^t$ are the same as those in Eq. (5).

Therefore, by accounting for the shilling attacker's level as well as the expected profit, the algorithm of VNS is as follows:

Input: $\boldsymbol{P} = p_j$: (n*1) item profit array, where $j \in [1,n]$;

$\boldsymbol{R_t} = r_{t,j}$: (1*n) user-item rating vector for an active user u_t;

$\boldsymbol{R_k} = r_{k,j}$:(m*n) user-item rating matrix for users whose ratings are used to make recommendations for active users, where $k \in [1,b]$; This set may include attacker profiles.

Procedures:

Step 1: Estimate the attacker probability (s_k) for each user (u_k) in $\boldsymbol{R_k}$ following the steps in Subsection 3.1.

Step 2: Retrieve the active user (u_t) profile in $\boldsymbol{R_t}$. Calculate the similarity $w(t,k)$ between u_t and each user (u_k) in $\boldsymbol{R_k}$ using Eq. (1).

Step 3: Find the optimal neighbor section decision $\boldsymbol{X}^t$ for u_t by solving the objective function in Eq. (7.1), with the constraints in Eqs. (7.2) and (7.3). The decision variable $\boldsymbol{X}^t = [x_1{}^t, x_2{}^t, \ldots x_k{}^t, \ldots x_m{}^t]$ denotes the neighbor selection list for the active user u_t, where $x_k{}^t = 1$ if the user u_k is selected as a neighbor and $x_k{}^t = 0$ if not.

$$Maximize\ SEP_t = \frac{\sum_{j=l}^{j=n} p_j * \sum_{k=1}^{k=m}(r_{k,j} - \bar{r}_k)*w(t,k)*x_k^t}{\sum_{k=1}^{k=m}|w(t,k)|*x_k^t} \quad (7.1)$$

$$s.t. \sum_{k=1}^{k=m} s_k * x_k^t \le \tau \quad (7.2)$$

$$\sum_{k=1}^{k=m} x_k^t \ge 1 \quad (7.3)$$

Output: $\boldsymbol{X}^t = x_k{}^t$: (1*m) binary neighbor selection vector for u_t, where $k \in [1,m]$.

This approach is designed to maximize the profits of an e-retailer when selecting neighbors to generate recommendations for each active user (Eq. 7.1), while maintaining the level of shilling attackers in the selected neighbors below the controlled level (Eq. 7.2). In addition, a constraint (Eq. 7.3) ensures that at least one neighbor is selected for each active user. Given that $x_k{}^t$ is a binary variable that indicates whether or not a user is selected, we can identify the potential undesirable outcome. That is, if a user's attack probability is higher than τ, the user is always excluded from the neighbor list.

4. Experimental evaluation

In this section we evaluate the performance of the proposed VNS method. We use a Book-Crossing [57] database consisting of approximately 433,671 explicit ratings provided by 77,805 users of 185,973 books. Ratings are expressed on a scale from 1 to 10. We reduced the original data set according to the number of ratings for each book (no < 5) and the rating frequency of each user (no < 20) to alleviate the sparseness in recommendation. The data set provided the ISBN of each book as well as other related information, such as title and author (s). We collected the posted price of the books from Amazon.com. Because of commercial confidentiality, we were not able to gather the actual cost of the books. Therefore, we followed two previous studies [28,58] to adopt the uniform distribution U (0.60, 0.80) of the posted price to estimate the cost of the book. We then calculated the profit as the difference between the price and the cost. We excluded the books for which we could not find price information. Finally, we obtained a subset of 2582 books, 733 users, and 30,061 ratings, with a mean rating value as 7.939.

To examine the performance of the proposed method, we performed five-fold cross-validation to prepare two sets of users: training and testing. In each fold, we considered every user in the testing set to be an active user. We selected each user's neighbors from users in the training set. For each user in the testing set, we also randomly withheld 50% of the user's book ratings as items to be recommended and used the remaining ratings to calculate similarity with users in the training set.

4.1. Experimental setting

To demonstrate the effectiveness of the proposed method, we compared the performance of the proposed VNS approach with the benchmark method. We simulated shilling attack profiles in various attack scenarios. For each training set, we simulated attack profiles using four types of attack models: random attack, average attack, bandwagon attack with RFM and bandwagon attack with AFM. For each attack model, we tried three different attack sizes {3%, 5%, and 10%} and seven different filler sizes {5%, 8%, 15%, 25%, 40%, 60%, and 85%}. For each combination of attack size and filler size, we merged its associated simulated attackers with users in the training set to create a new training set, then tested the performance of the method on the testing set. In the first set of simulations, only one item in the attack profiles is selected as the target. The chosen item should be an unpopular item with an average rating below the global mean rating and for which the number of ratings is relatively smaller than that of other items. We set the size of the selected items as 1% for bandwagon attacks. Whether the method effectiveness is affected by additional target items or more selected items is evaluated in robustness test in Subsection 4.3.

We first evaluated the ability of the proposed method to control the influence of attackers. We used *mean absolute error* (MAE) to evaluate the effectiveness on preserving rating prediction accuracy as:

$$MAE = \frac{1}{\sum_{k \in U_T} ||I(k)||} \sum_{k \in U_T} \sum_{j \in I(k)} |r_{k,j} - \hat{r}_{k,j}| \quad (8)$$

where $r_{k,j}$ is the actual rating of the user k on item j; $\hat{r}_{i,j}$ is the predicted rating; $I(k)$ is the set of recommended items for the user k; and U_T is the set of all the users in the testing set.

Because our proposed method considers profit in neighbor selection, we also evaluated methods using *mean similarity-based expected profit* (MSEP). For each user in the testing set, we calculated SEP using Eq. (6). The MSEP of each testing set is the average SEP of all its users, defined as:

$$MSEP = \frac{\sum_{t \in U_T} SEP_t}{||U_T||} \quad (9)$$

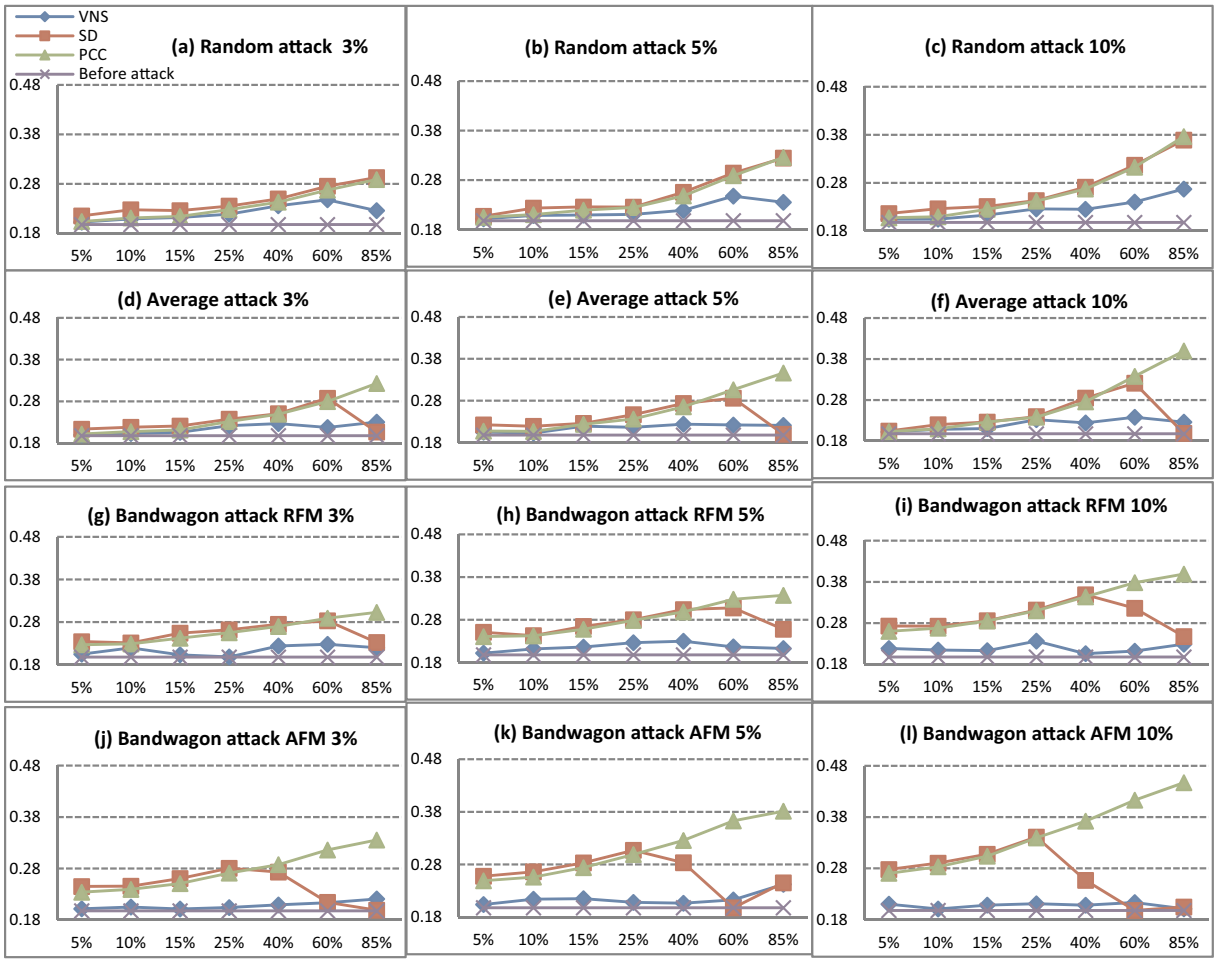

Fig. 2. Comparative Results for MAE.

where SEP_t is defined in Eq. (6), and U_T represents all active users in the testing set.

We evaluated the performance of the proposed method using two benchmarks. The first is the original recommender systems without attack detection, that is, the traditional Pearson Correlation Coefficient (PCC). The second is a well-known attack detection benchmark: Lee and Zhu's shilling attack detection (SD) method [18]. By using these two benchmarks, we were able to assess whether our proposed method preserved recommendation quality and whether it was superior to other methods. For each new training set (i.e., the training set with attack profiles), we employed VNS and SD respectively to (1) detect attackers, (2) select proper neighbors for users in the testing set, and (3) use the selected neighbors to predict the ratings in the testing set. With regard to PCC, since it does not discriminate attackers from genuine users, no attack detection occurred in the training set. The shilling attacker level threshold τ was kept at 0.5 in the implementation of VNS. We discuss the impact of the value of τ on the performance in Subsection 4.3.

4.2. Evaluation results

Figs. 2 and 3 summarize the experimental results from cross-validation. Fig. 2 reports the comparative results corresponding to MAE for our proposed method and the benchmark methods in 84 attack scenarios (e.g., four attack models with three attack sizes and seven filler sizes). Fig. 3 reports the comparative results corresponding to MSEP. To better evaluate prediction performance, we also calculated the MAE of the recommender system "before attack," that is, the average difference between the actual ratings and the predicted ratings without attackers. This value measures the prediction error irrelevant to shilling attacks, but from the user-based collaborative filtering recommender system itself. It provides a baseline to assess the performances of the proposed method and the benchmark method in controlling influence from shilling attacks.

Because PCC does not exclude attackers in its rating predictions, we referred to the performance of PCC to explore the impact of various attack scenarios on prediction quality. Fig. 2 shows that its MAE is always larger than that of the system "before attack." Moreover, for all types of attack models, the MAE of PCC increases with attack size and filler size. This result indicates that adding more attackers or filler items leads to more serious attacks with more biased ratings. In addition, compared with other attack models, random attacks are less of a threat to recommender systems, because they have smaller impacts on prediction quality (e.g., smaller MAEs). Fig. 2 shows that our proposed method is more advantageous than the benchmark methods for preserving prediction accuracy. Compared with the traditional PCC method, the proposed method is advantageous in every attack scenario in terms of MAE. Compared with Lee and Zhu's shilling attack detection method, the proposed method has a lower prediction error in every attack scenario that adopts the random filling method (i.e., random attacks and bandwagon attacks with RFM). When the attack model uses the average filling method, the proposed method performs better than SD in all scenarios except those with very large filler sizes (60% or 85%). Fig. 3 shows that our proposed method yields consistently larger expected profits than the benchmark methods in all attack scenarios. The MSEP advantage of our proposed method is as expected since neither benchmark method takes profit into consideration. Thus,

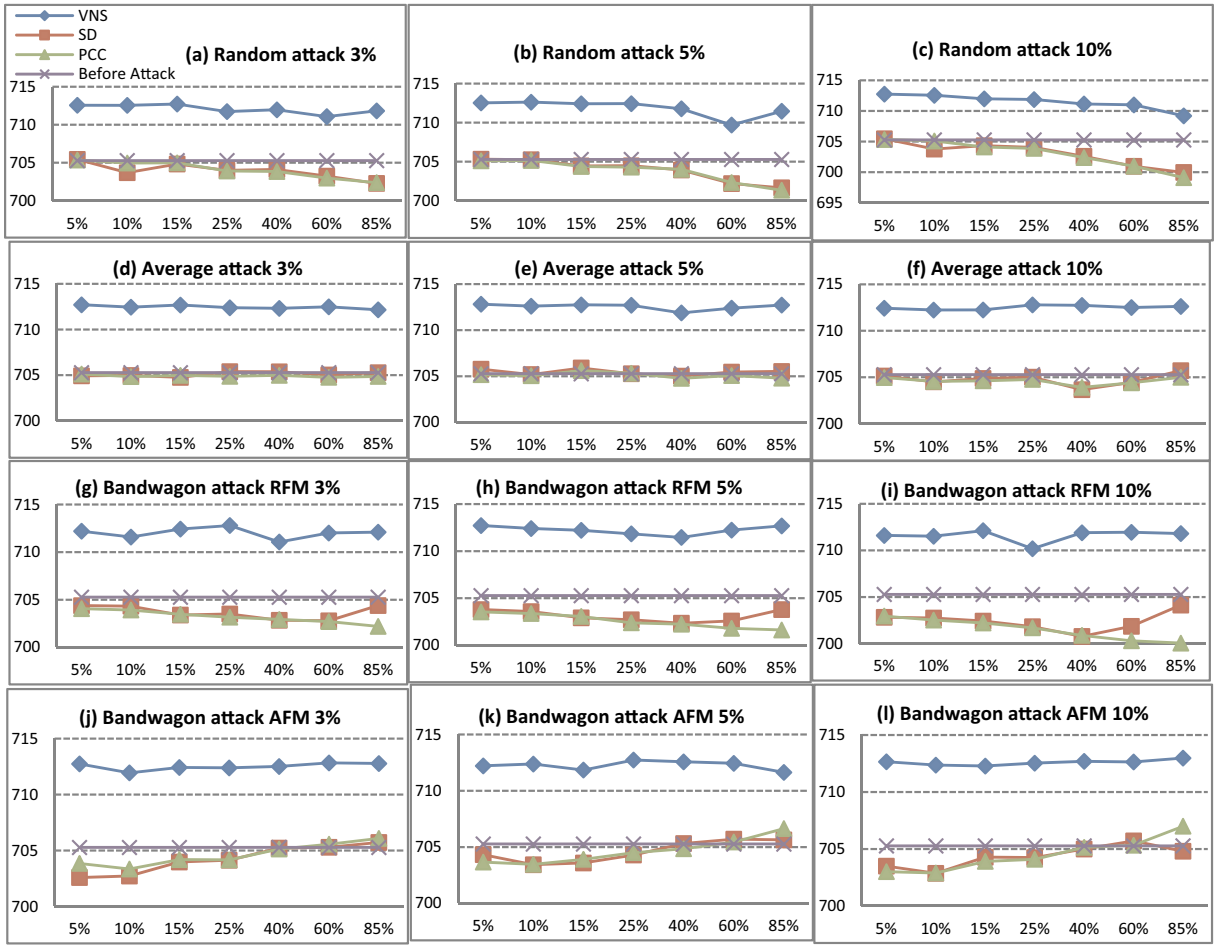

Fig. 3. Comparative Results for MSEP.

experimental results show that compared with the benchmark methods, our proposed model increases profits while maintaining prediction accuracy.

Another interesting finding of our experiments is that applying the shilling attack detection method may not necessarily help improve recommendation quality. The MAE average of SD across all types of attacks is 0.256. Although this average is better than that of PCC (0.276), the MAE of SD is not advantageous in every attack scenario. According to Fig. 2, in cases of random attacks, SD, which is an effective attack detection method, is slightly worse than PCC, which does not discriminate shilling attackers at all, in terms of prediction accuracy. For random attacks, the MAE average of SD across all filler sizes and attack sizes is 0.254, whereas that of PCC is 0.248. In cases of average attacks, bandwagon attacks with RFM and bandwagon attacks with AFM, when filler sizes are relatively small (between 5% and 40%), the MAEs of SD also are slightly worse than those of PCC. When filler sizes are between 5% and 40%, the MAE average of SD across all three attack models and all three attack sizes is 0.261, and that of PCC is 0.259. However, with increased filler size, the attack detection method shows a clear advantage, especially when the attack size is large. In average and bandwagon attacks, the prediction accuracy of PCC decreases largely because of the malicious ratings injected by attackers. When filler sizes are large (60% or 85%), PCC 's MAE average across attack models and attack sizes is 0.349, whereas in those attack scenarios, SD discriminates attacks successfully, such that its MAE average is much smaller (0.245). Notably, for every attack size with average and bandwagon attacks, because of its advantage in cases of large filler sizes, Lee and Zhu's shilling attack method always has a smaller average prediction error across all filler sizes than the traditional PCC.

The above experimental results show that though random attacks or attacks with smaller filler sizes are relatively less of a threat to recommender systems, applying the traditional filtering-based detection strategy (e.g., SD) in such attack scenarios would actually further diminishes the prediction accuracy of recommender systems (compared to the no-detection method PCC). But applying the proposed method in these attack scenarios can help improve prediction accuracy, because in such attack scenarios, SD suffers from high false-positive rates and low true-positive rates of attack detection. It is likely to misidentify genuine users as attackers. Because it gives each user a binary decision, it filters users out of the system once they are identified as attackers. Thus, in such attack scenarios, the rating prediction error of SD becomes large, not only because of the biased ratings injected from attackers in the system, but also because of losing the useful preference information from genuine users who are mistakenly removed from the system. In contrast, because PCC does not filter out any attackers, all genuine users remain in the system, and their information is preserved. Therefore, in such attack scenarios, PCC can outperform SD in terms of prediction accuracy. Our proposed method, however, uses a probability other than a binary decision for suspicious users. Thus, it can both undermine the impacts of the correctly identified attackers and partially reserve the rating preference information carried from genuine users, even when they are assigned attack probabilities. Hereby, it maintains better prediction quality than the other methods in such attack scenarios.

In attack scenarios that pose more serious threats to recommender systems—for example, average attacks with AFM or attacks with large filler sizes—attackers tend to be gathered at the center of all users such that they can more effectively influence the predicted ratings to genuine users [18]. At the same time, they are more likely to be detected

using the feature of their own distance from other users. In such attack scenarios, both Lee and Zhu's shilling attack method (SD) and our proposed method can successfully mitigate the impacts from attackers, such that they outperform PCC in terms of prediction accuracy. In these attack scenarios, our proposed method still outperforms SD in terms of prediction accuracy, except in a couple of attack scenarios in which SD keeps genuine users in the system and filters out attackers accurately. Therefore, the experimental results in all types of attack scenarios confirm that our proposed method is advantageous for both prediction accuracy and expected profit.

4.3. Robustness tests

We conduct additional simulations to evaluate the robustness of our method under various attack conditions.

4.3.1. Simulation parameters

As mentioned in Subsection 4.1, we used fixed values in some parameters when simulating the attacks. We evaluated whether the performance advantage of our proposed method was limited to those specific parameter values. First, we assessed whether using multiple target items would affect the performance of the methods. We simulated attacks with three target items in 84 attack scenarios as in 4.1. Similar to the selection criteria used in Subsection 4.1, all chosen target items were unpopular. In this evaluation, we used the experimental results for bandwagon attacks with AFM as an illustration to report the performance results for two reasons: (1) The benchmark method SD has a relative advantage in this attack model, in that there are more attack scenarios from bandwagon attacks with AFM than from the other three models in which SD achieves the lowest prediction error (e.g., 6 of 21 attack scenarios in Fig. 2), and (2) The performance difference of each method between single-target and multiple-target attacks is qualitatively the same among the four attack models. The detailed results for the other three attack models are available on request. Fig. 4 shows the comparative results corresponding to both MAE and MSEP. When we compare the MAEs of PCC in Fig. 4 with those in Fig. 2, we see that the increase in target items does not have much effect on the prediction deviation. Experimental results show that the findings from Subsection 4.2 remain valid. The advantage of VNS over the benchmark methods expands with the increased number of target items; It outperforms with regard to both expected profit and the accuracy in almost all attack scenarios. The only exception occurs when the filler size is 0.6; consistent with the previous result, SD continues to show the smallest prediction error. Thus, the increased number of target items does not affect the advantage of our method in its performance effectiveness.

Next, we tested the impact of the selected item size in bandwagon attacks. We simulated bandwagon attacks using both the RFM and the AFM with an increased selected item size as 2%. Again, we used the results of bandwagon attacks with AFM to illustrate performance differences, which are qualitatively similar to those of bandwagon attacks with RFM. Fig. 5 shows the experimental results. A comparison of Fig. 5 to Fig. 2, showing the difference in MAEs of PCC, indicates that the prediction deviation of the recommender system increases largely with the increase of selected items. Attacks with larger selected item sizes pose greater threats to recommender systems, in which SD would have the advantage in attack detection. When the selected item size is larger, the prediction accuracy advantage of SD over PCC increases; SD outperforms PCC in terms of prediction accuracy even in cases of moderate filler size (40%). Nevertheless, VNS maintains a smaller prediction error than SD except when filler sizes are large (60% or 85%). In addition, the advantage of the proposed method over the traditional PCC in terms of prediction accuracy and that over both benchmark methods in terms of expected profit remain the same. Therefore, the performance effectiveness of the proposed method remains valid with a different selected item size.

4.3.2. Shilling attacker level

After choosing the threshold of the shilling attacker level τ at 0.5 in the implementation of VNS (Subsection 4.1), we evaluated whether its effectiveness was affected by the threshold value. We examined its performance with two larger shilling attacker levels {i.e., 1, 2}. We did not test the performance with a smaller τ, because it measured the expected number of attackers, and the current number was not relatively high. We used the same attack files simulated in 4.1, applying VNS to select appropriate neighbors with the new τ, using the selected neighbors to make predictions, and recording MAEs and MSEPs respectively. Fig. 6 compares the performances among four methods: PCC, SD, VNS with τ as 1, and VNS with τ as 2. According to the considerations as in Section 4.1, we only present the experimental results for bandwagon attack with AFM as an illustration of the performance results. Fig. 6 shows that because the attack files are the same as those in Section 4.1, the performance results of both benchmark methods are the same as those for bandwagon attacks with AFM in Figs. 2 and 3. According to Fig. 6, a larger value of τ would decrease the prediction accuracy of VNS, because more suspicious users would be included in the neighbor list. It also shows that VNS with a τ value of 2 has a larger prediction error than those with τ values of 1 or 0.5. Nevertheless, its performance advantages over the two benchmark methods, in terms of both MAE and MSEP, remain consistent.

5. Evaluation with the profit-based benchmark method

The benchmark methods that we used in the previous section focus only on shilling attack detection; it does not consider the profit

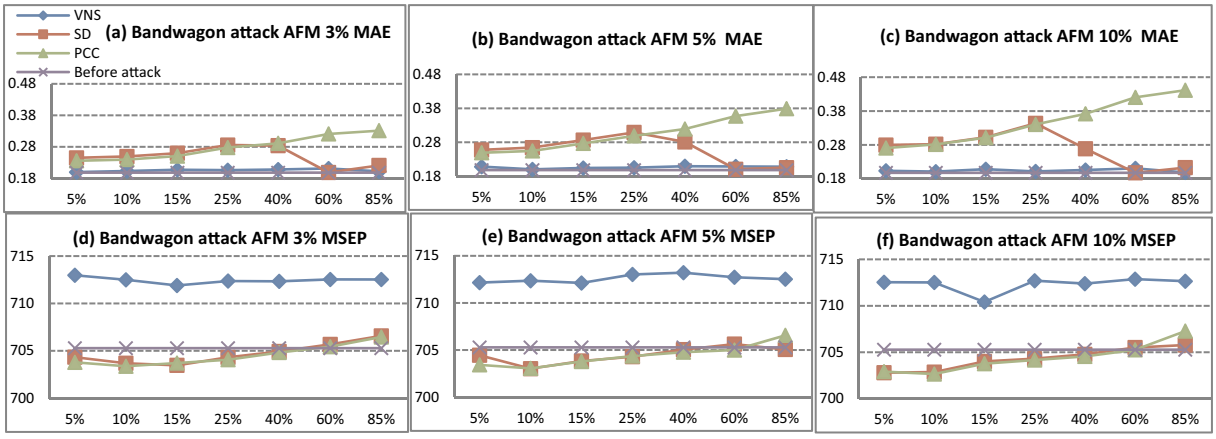

Fig. 4. Comparative results, three target items, bandwagon attack with AFM.

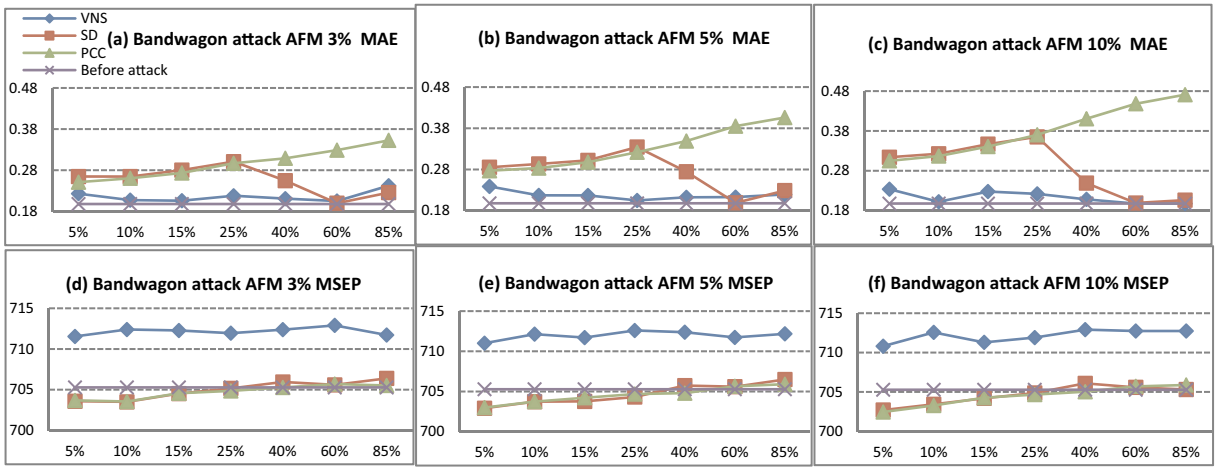

Fig. 5. Comparative results, selected item size 2%, bandwagon attack with AFM.

associated with recommendations. In this section, we compare VNS with the profit-based benchmark method, HPRS [24]. The key idea of HPRS is that it recommends the top K items with the largest similarity-based profit (i.e., item profit × similarity-based purchase probability). Another method for profit-based recommendation is CPPRS [24]. We selected the HPRS instead of the CPPRS as the benchmark, because CPPRS was designed for recommendations based on users' purchase frequency, which did not apply to the current problem. We used the same experimental settings as in Section 4.1, with four different attack models, three attack sizes, and seven filler sizes. HPRS does not deal with the influence of attacks, so it is vulnerable to misleading ratings. To better illustrate the effectiveness of our proposed method, we compared VNS with an integrated approach (HPRS + SD), first applying SD to filter out attackers and then using HPRS to make recommendations. Because the output of HPRS was a list of recommended items rather than ratings, we were not able to use MAE to compare performance. Instead of calculating the deviation between the predicted rating and the actual rating, we evaluated performance by comparing the recommended items against users' actual preferences. In [24], each item to be recommended would be assigned a binary decision, i.e., recommended or not recommended. Thus, we discretized each item to be recommended as "liked" if its rating in the original data set was 10 and "disliked" otherwise, thereby comparing the recommended items with the "liked" items to assess whether they reflected active users' actual preferred items. We adopted the same metrics as used in [24], precision, recall, and F1, to evaluate the prediction accuracy of the methods, as shown in Eqs. (10)-(12).

$$Precision = \frac{True\ positive}{True\ positive + False\ positive} \tag{10}$$

$$Recall = \frac{True\ positive}{True\ positive + False\ negative} \tag{11}$$

$$F_{measure} = \frac{2 \bullet Precision \bullet Recall}{Precision + Recall} \tag{12}$$

True positive indicates the number of correctly recommended items, and *False positive* is the number of "disliked" items incorrectly recommended; *False negative* is the number of real "liked" items that failed to be recommended. In the implementation of VNS, instead of predicting items' ratings solely, we also selected the top K recommended items. Specifically, for each user in the testing sets, we first used the neighbors selected from VNS to predict ratings and then selected the top K items with the highest ratings. When comparing the methods in terms of the recommendation profitability, we used a measure similar to that used in [24], the *mean of total profit* (MTP) from recommendations. This measure differs from MSEP and is the average of all active users' total profits associated with their top K recommended items.

Table 2 shows the comparative results between the proposed method and (HPRS + SD) with four attack models and three attack

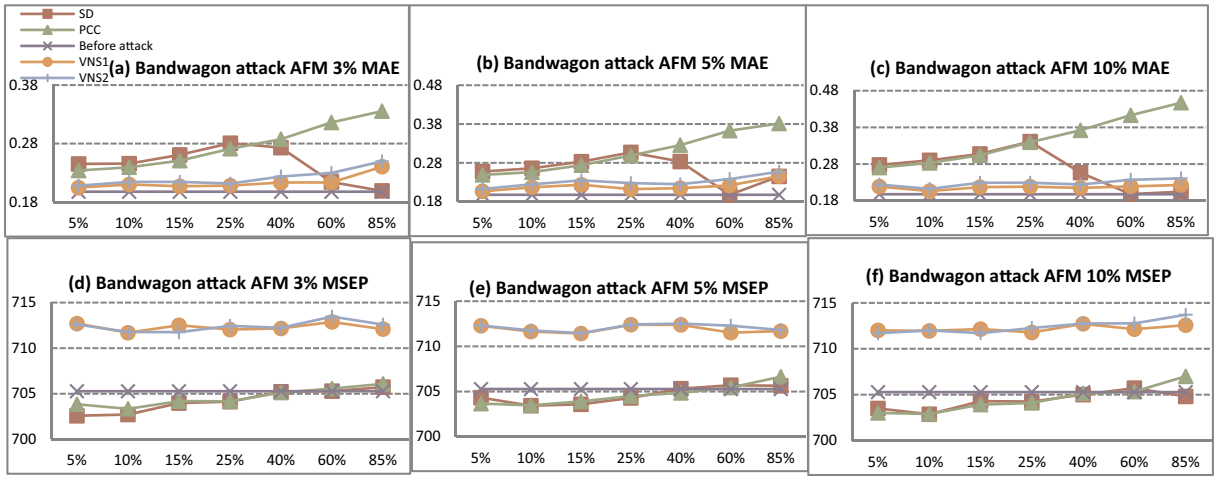

Fig. 6. Comparative Results with Two Tau Values, Bandwagon Attack with AFM
Note. VNS1 represents the result for VNS with tau = 1 & VNS2 represents the result for VNS with tau = 2.

Table 2
Comparative Results between VNS and (HPRS + SD) method.

Measure method		Random attack			Average attack			Bandwagon attack with AFM			Bandwagon attack with RFM		
		Attack size (%)											
		3	5	10	3	5	10	3	5	10	3	5	10
Precision	VNS	0.750	0.747	0.744	0.751	0.750	0.749	0.753	0.750	0.751	0.750	0.746	0.748
	HPRS + SD	0.498	0.498	0.497	0.501	0.498	0.499	0.498	0.495	0.497	0.497	0.495	0.494
Recall	VNS	0.718	0.716	0.714	0.724	0.720	0.721	0.725	0.724	0.725	0.723	0.719	0.719
	HPRS + SD	0.431	0.430	0.429	0.433	0.431	0.435	0.430	0.430	0.430	0.426	0.426	0.424
F1	VNS	0.615	0.613	0.610	0.618	0.616	0.616	0.620	0.618	0.619	0.617	0.613	0.614
	HPRS + SD	0.388	0.388	0.386	0.390	0.388	0.390	0.388	0.386	0.387	0.386	0.385	0.384
MTP	VNS	26.773	26.737	26.666	26.758	26.822	26.874	26.818	26.812	26.825	26.743	26.794	26.678
K = 5	HPRS + SD	32.680	32.717	32.710	32.692	32.711	32.698	32.670	32.723	32.724	32.724	32.744	32.757

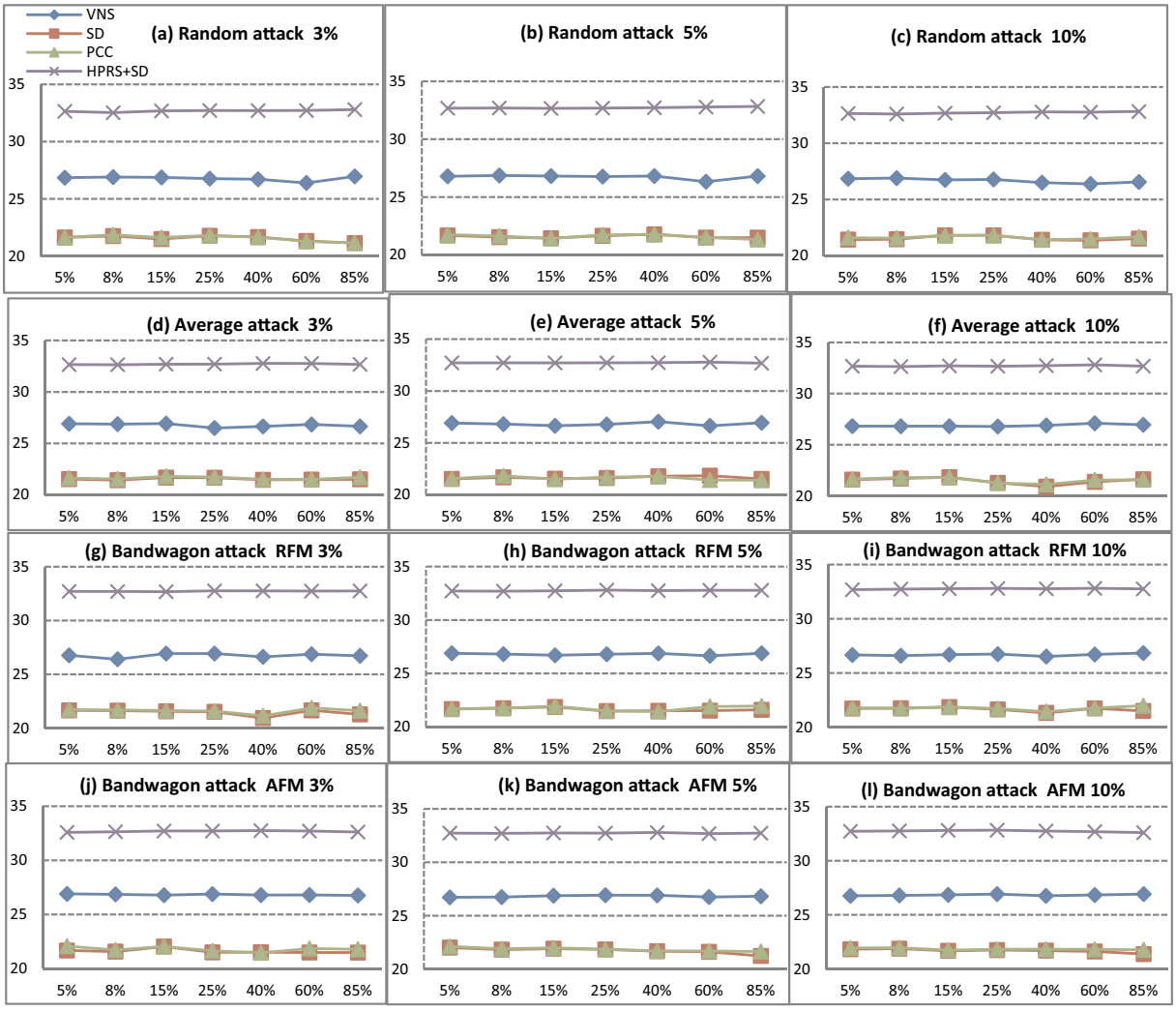

Fig. 7. Comparative results for MTP with the top five recommended items.

sizes. We calculated MTP according to recommendations for five items. The overall precision and recall values are affected by the value of K. For an active user with more "liked" items, a larger K results in more genuine "liked" items being accurately recommended (i.e., smaller false negative and increased recall). For an active user with fewer "liked" items, a larger K results in more falsely "liked" items being recommended (i.e., larger false positive and decreased precision). Thus, we selected $K = 5$ as the average number of the "liked" items is 5.061. Table 2 shows the averages of each measure across seven filler sizes {5%, 8%, 15%, 25%, 40%, 60%, and 85%}. Fig. 7 provides more detailed information, showing the MTPs of the top five recommended items at each filler size. We also included SD and PCC as benchmarks and calculated their respective MTPs. Unlike MSEP, MTP measures the profit from top K items. The performance difference in terms of MTP may not be consistent with that of MSEP. We do not include PCC or SD results in Table 2, since we have already evaluated them in terms of MAE in Subsection 4.2. Both MAE and precision/recall capture the prediction accuracy. We expect the performance differences among methods using precision/recall to be qualitatively consistent with those using MAE.

Fig. 7 shows that in every attack scenario, the total profit of the recommended items using the proposed method is smaller than that using (HPRS + SD). However, it is much larger than that using PCC or SD. This result is as expected, given that (HPRS + SD) always recommends the most profitable items. However, the high profit from (HPRS + SD) is realized at a cost of its prediction accuracy. Though (HPRS + SD) uses SD to mitigate the influence from attacks and also considers the prediction accuracy by using the purchase probability, Table 2 shows that its prediction accuracy is significantly worse than VNS regardless of the attack scenario. On average, the overall prediction accuracy of VNS drops 37.128% compared with that of (HPRS + SD) in terms of F1. Therefore, though the profit associated with our proposed method is not as large as the pure profit-driven method, it achieves a balance between satisfying users' expected satisfaction and increasing profitability.

6. Evaluation with a mixture of attack models

The evaluations in previous sections deal with only one type of attack model at a time. In this section, we evaluate whether the effectiveness of our proposed method remains valid with a mixture of attack types in a system. For each training set, we simulated attack profiles using four types of attack models: random attack, average attack, bandwagon attack with RFM and bandwagon attack with AFM, each with an attack size of 1% or 2%. We merged the attack profiles of various attack models with the same attack size and created new attack profiles with an attack size of 4% or 8%. For each attack size (4% or 8%), we simulated attack profiles using seven filler sizes {5%, 8%, 15%, 25%, 40%, 60%, and 85%} as in 4.1.

Fig. 8(a–b) and (d–e) show MAEs and MSEPs of VNS, PCC and SD with attack sizes of 4% and 8% respectively. According to MAEs of PCCs, in the mixed attack, both larger filler sizes and larger attack sizes create more challenges to recommender systems. There are two major findings: First, the threat level from a mixed attack falls between those of random attack and bandwagon attack with AFM. For example, if we compare Figs. 8(a) and 8(d) with Fig. 2, we see that the rating deviation of PCC in the mixed attack at the attack size 4% is smaller than that in the bandwagon attack with AFM at the attack size of 3%, but greater than that in the random attack at the attack size of 5%. Second, the prediction accuracy of VNS is better in a single type of attack than in a mixed attack. The average MAE of VNS in the mixed attack across all attack sizes and filler sizes (0.235) is larger than that in the random attack (0.222), which is the largest among the four attack types. However, VNS retains its advantage over the two benchmark methods in the mixed attack in terms of both MAE and MSEP; VNS outperforms the benchmark regardless of attack and filler sizes. Compared with that of PCC (0.289), the average MAE of SD across all attack sizes and filler sizes is slightly better (0.282). But similar to the findings in Subsection 4.2, the accuracy advantage of SD begins to show only when the filler size is larger than 40%. Hence, the mixed attack model brings more challenges to the effectiveness of VNS, but its advantage over the benchmark methods remains.

Next, we compared the performance of VNS with the profit-based model (HPRS + SD) in terms of profitability and prediction accuracy of the mixed attack model. Following the same experimental procedures used in Section 5, we applied each method respectively to the previous simulated attack profiles to obtain the top five recommended items. We adopted the measures used in Section 5. Table 3 summarizes the experimental results related to prediction accuracy; it shows comparative results corresponding to precision, recall, and F1 between VNS and the benchmark method (HPRS + SD) at seven different filler sizes (in the seven columns). Fig. 8(c) and 8(f) show comparative results corresponding to the MTPs of the top five recommended items between our proposed method and the three benchmark methods (i.e., HPRS + SD, PCC, and SD) at two attack sizes and seven different filler sizes. We include PCC and SD in the comparison of profitability, according to the same consideration as those in Section 5.

The profit associated with the top five items recommended resulting from VNS is smaller than that from (HPRS + SD) but always larger than that from PCC or SD in any attack scenario. Although (HPRS + SD) shows an obvious profitability advantage, its prediction accuracy is far smaller than that of VNS in terms of precision, recall, and F1. Therefore, VNS is advantageous over the benchmark methods for increasing profit and maintaining prediction accuracy, even in more complicated attack scenarios such as the mixed attack model.

7. Discussion and conclusion

In this study, we addressed an unsettled but important issue associated with user-based collaborative filtering recommender systems: how can we redesign recommender systems under shilling attacks to ensure that they provide recommendations that balance prediction

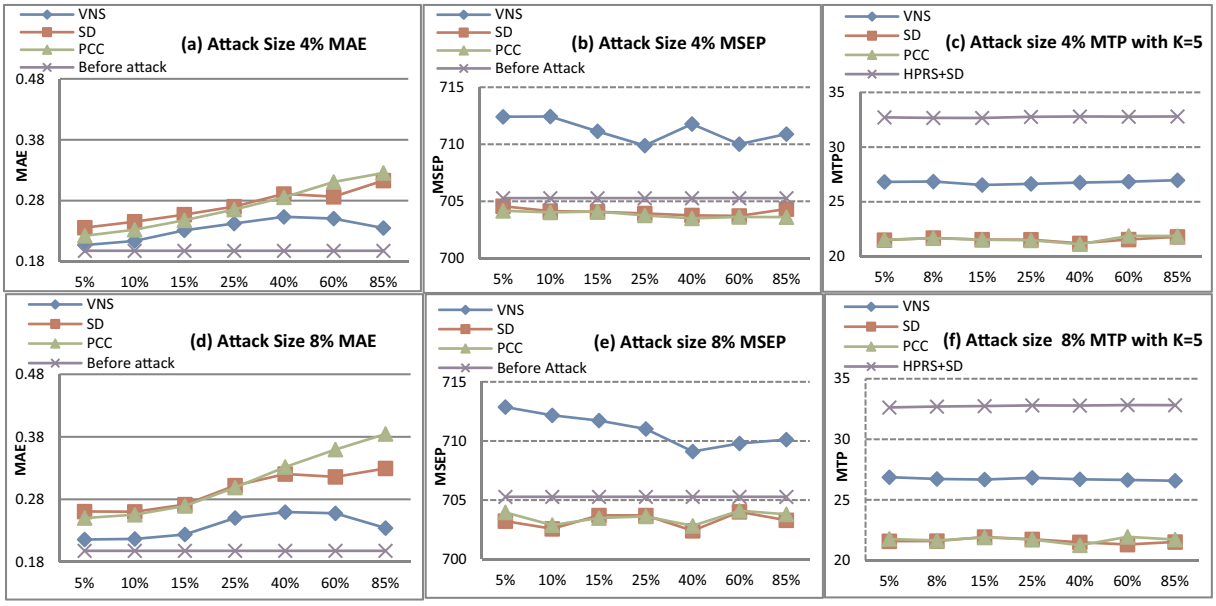

Fig. 8. Comparative results in the mixture attack.

Table 3
Comparative results between VNS and (HPRS + SD) method in the mixture attack.

Attack size (%)	Measure	Method	Filler size (%)						
			5	8	15	25	40	60	85
4	Precision	VNS	0.753	0.753	0.745	0.720	0.720	0.690	0.714
		HPRS + SD	0.494	0.500	0.504	0.498	0.488	0.494	0.488
	Recall	VNS	0.724	0.726	0.720	0.683	0.692	0.650	0.689
		HPRS + SD	0.419	0.426	0.438	0.431	0.426	0.427	0.418
	F1	VNS	0.619	0.620	0.614	0.585	0.588	0.558	0.585
		HPRS + SD	0.382	0.388	0.393	0.387	0.380	0.386	0.379
8	Precision	VNS	0.755	0.745	0.747	0.741	0.690	0.688	0.700
		HPRS + SD	0.500	0.494	0.500	0.490	0.488	0.484	0.486
	Recall	VNS	0.728	0.715	0.719	0.709	0.656	0.655	0.667
		HPRS + SD	0.432	0.422	0.434	0.414	0.417	0.414	0.412
	F1	VNS	0.622	0.611	0.613	0.607	0.558	0.557	0.570
		HPRS + SD	0.388	0.383	0.389	0.378	0.379	0.376	0.376

accuracy and e-retailer profits? The motivation for solving this problem is that shilling attacks are becoming more prevalent in real-world. To date, studies have considered the redesign of recommender systems under shilling attacks only in terms of prediction accuracy following detection. However, real-world experiences suggest that quality recommendations should consider not only accuracy but also profitability. Accordingly, we addressed this challenge by taking both e-retailer profits and the influence of shilling attacks into account.

We proposed a new method known as VNS; with this method, for each active user we selected an optimal list of neighbors to make recommendations. This approach increased the expected profit gain of e-retailers while protecting the recommendations from shilling attacks. To validate the performance of the proposed method, we conducted several experiments and compared our approach with various benchmarks. Experimental results show that VNS yields higher prediction accuracy and profits than previous shilling detection methods in various attack scenarios. Compared with the profit-based benchmark method, it is advantageous in that it balances accuracy and profit.

This study integrates the profitability issue into recommender systems that are under attack. Our proposed method has practical implications for e-retailers' deployment of recommender systems. The importance of the value-dimension of recommendation is not fully recognized in prior studies on recommender systems, which limit their potential applicability in real-world scenarios. The experimental results of this study demonstrated the advantages of the proposed method in both accuracy and profitability. Hence, it is a feasible approach adopted by e-retailers in the real-world. The aim of the proposed method is to realize sustainable profitability, without just recommending the most profitable items. It also recognizes the role of recommendation relevancy in maintaining customer's loyalty. Thus, from the customer perspective, recommendations resulting from our proposed method are more relevant to customers' actual preferences—they resist attackers' influence and thereby enhance customer satisfaction. From the e-retailer perspective, such recommendations not only increase profits but also help retain customers. Furthermore, experimental results demonstrate that it is an effective approach to protect recommender systems from various attackers' behavior, making it more easily implemented by e-retailers. For example, some real-world attackers may be persistent—they keep adding filler items or selected items to the system so as to accumulate their influence. e-Retailers can use the proposed method to limit the influence from such persistent attackers, since its robustness is not affected by the filler or the selected item size. Besides, in a real-world scenario, some strategic attackers may adopt multiple attack models to realize their attack intention. As shown in the experimental results, e-retailers still can rely on the proposed method to counter attacks not only in a single model, but also in such a complicated situation.

In terms of methodology, this study contributes to literature on recommender systems primarily in three ways. First, the study fills the research gap in recommender systems by investigating how to maintain e-retailers' profitability in the shilling attack context using a novel approach. e-Retailer profitability and shilling attacks are two critical challenges in the design of recommender systems. They are usually handled separately in prior literature, which limits the applications of the previously proposed methods, while they are both addressed in this study. Second, the study proposes a different approach to limit the influence of attackers. Previous shilling attack detection strategies tend to suffer from high false-positive rates. In this study, we leverage the features of shilling attackers to estimate the attacker probability of each user. Empirical results show that this approach overcomes the limitations of previous filtering-based approaches; it is more effective in preserving prediction accuracy amidst various attack scenarios. Third, it shows that profit can be realized from the strategic selection of neighbors. Other profit-based recommender systems focus on direct item selection; such a strategy can realize the highest possible profit, though usually at the cost of prediction accuracy. Our proposed strategy increases profit through neighbor selection, while better preserving accuracy, because recommendations are still made according to predicted preferences.

This study has several limitations that should be addressed in future research. Our methodology is designed according to the user-based collaborative filtering algorithm. Some recommender systems (e.g., content-based) [4] do not rely on others' opinions to make predictions. Although our proposed neighbor selection method may not be generalizable to these recommender contexts, they are less vulnerable to shilling attacks. Moreover, we did not evaluate the performance of our proposed method in the case of a segment attack. It is possible that the effectiveness of our method is limited in this type of attack, because it uses the minimum rating in the filler items. The features of shilling attackers that we use in this study, such as high similarity with the majority of the users, may not be applicable. However, because of segment attackers' extremely unusual ratings, they can be easily detected using rating deviation in small to medium attack sizes. Furthermore, our data suggest another deficiency: We calculated expected profit from recommendations using predicted preferences of every active user. This method implicitly assumes that every active user reacts to the recommended items in the same way, which may not be the case in reality. Some users may be more sensitive than others to recommendations [59]. Therefore, sensitivity toward recommendations should be included in the calculation of expected profits. In addition, the data lacks users' demographic information, which helps analyze users' similarity [60]. Demographic data could also alleviate the cold start problem, another data integrity issue in recommender systems, which is not addressed specifically in this study [5,60]. In the future, the proposed method could be expanded to incorporate new information and solve other integrity issues of recommender systems.

Acknowledgement

We would like to thank the Editor-in-Chief (James R Marsden) and two anonymous reviewers who provided helpful guidance throughout the review process. Support for this project was provided by a PSC-CUNY Award, jointly funded by The Professional Staff Congress and The City University of New York.

References

[1] J. Schafer, J. Konstan, J. Riedl, E-commerce recommendation applications, Data Mining Knowledge Discovery 5 (2001) 115–153.
[2] K. Cheung, J.T. Kwok, M.H. Law, K. Tsui, Mining customer product ratings for personalized marketing, Decision Support Systems 35 (2003) 231–243.
[3] B. Xiao, Benbasat. I. E-commerce product recommendation agents: use, characteristics, and impact, MIS Quarterly 31 (1) (2007) 137–209.
[4] A. Mild, T. Reutterer, An improved collaborative filtering approach for predicting cross category purchases based on binary market basket data, Journal of Retailing and Consumer Services 10 (2003) 123–133.
[5] C.Y. Chung, P.Y. Hsu, S.H. Huang, βp: a novel approach to filter out malicious rating profiles from recommender systems, Decision Support Systems 55 (2013) 314–325.
[6] S. Bag, S. Kumar, A. Awasthi, M.K. Tiwari, A noise correction-based approach to support a recommender system in a highly sparse rating environment, Decision Support Systems 118 (2019) 46–57.
[7] J.S. Breese, D. Heckerman, C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, 1998, pp. 43–52.
[8] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, Grouplens: an open architecture for collaborative filtering of netnews, Proceedings of Conference on Computer Supported Cooperative Work, CA (1994) 219–231.
[9] S. Lam, J. Riedl, Shilling recommender Systems for fun and Profit, 13th International World Wide Web Conference, ACM, New York, 2004, pp. 309–402.
[10] M. O'Mahony, N. Hurley, N. Kushmerick, G. Silvestre, Collaborative recommendation: a robustness analysis, ACM Transactions on Internet Technology 4 (4) (2004) 344–377.
[11] P.A. Chirita, W. Nejdl, C. Zamfir, Preventing shilling attacks in online recommender systems, Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management, NY, 2005, pp. 67–74.
[12] B. Mobasher, R. Burke, R. Bhaumik, C. Williams, Toward trustworthy recommender systems: an analysis of attack models and algorithm robustness, ACM Transactions on Internet Technology 7 (2007) 1–40.
[13] M.P. O'Mahony, N.J. Hurley, G. Silvestre, Detecting noise in recommender system databases, Proceedings of the 11th International Conference on Intelligent User Interfaces, 2006, pp. 109–115.
[14] S. Zhang, A. Chakrabarti, J. Ford, F. Makedon, Attack Detection in Time Series for Recommender Systems. Proceeding of 12th ACM SIGKDD, ACM, New York, 2006, pp. 809–814.
[15] B. Mehta, T. Hofmann, P. Fankhauser, Lies and propaganda: Detecting spam users in collaborative filtering, 12th International Conference on Intelligent User Interfaces, ACM, New York, 2007, pp. 14–21.
[16] C. Williams, B. Mobasher, R, Burke. Defending recommender systems: detection of profile injection attacks. Service Oriented Computing and Applications, 1, (2007), 157–170.
[17] R. Bhaumik, B. Mobasher, R.D. Burke, A clustering approach to unsupervised attack detection in collaborative recommender systems, 7th IEEE International Conference on Data Mining, Las Vegas, NV, 2011, pp. 181–187.
[18] J. Lee, D. Zhu, Shilling attack detection—a new approach for a trustworthy recommender system, INFORMS Journal of Computing 24 (1) (2012) 117–131.
[19] F. Zhang, Q. Zhou, HHT-SVM: an online method for detecting profile injection attacks in collaborative recommender systems, Knowledge Based System 65 (2014) 96–105.
[20] H. Xia, B. Fang, M. Gao, H. Ma, Y. Tang, J. Wen, A novel item anomaly detection approach against shilling attacks in collaborative recommendation systems using the dynamic time interval segmentation technique, Information Sciences 306 (2015) 150–165.
[21] Z. Yang, Z. Cai, X. Guan, Estimating user behavior toward detecting anomalous ratings in rating system, Knowledge Based System 111 (2016) 144–158.
[22] A. Das, C. Mathieu, D. Ricketts, Maximizing Profit Using Recommender Systems, WWW, NC, 2010.
[23] W. Shih, S. Kaufman, D. Spinola, Netflix. Case, Harvard Business School, Boston, MA, 2007.
[24] L. Chen, F. Hsu, M. Chen, Y. H, Developing recommender systems with the consideration of product profitability for sellers, Information Sciences 178 (2008) 1032–1048.
[25] U. Panniello, S. Hill, M. Gorgoglione, The impact of profit incentives on the relevance of online recommendations, Electronic Commerce Research and Applications 20 (2016) 87–104.
[26] J. Castro, R. Yera, L. Martínez, An empirical study of natural noise management in group recommendation systems, Decision Support Systems 94 (2017) 1–11.
[27] Y. Jiang, J. Shang, Y. Liu, Maximizing customer satisfaction through an online recommendation system: a novel associative classification model, Decision Support Systems 48 (3) (2010) 470–479.
[28] Y. Jiang, J. Shang, C.F. Kemerer, Y. Liu, Optimizing E-tailer profits and customer savings: pricing multistage customized online bundles, Marketing Science 30 (4) (2011) 737–752.
[29] G. Shani, R. Brafman, D. Heckerman, An MDP-based recommender system, Journal of Machine Learning Research 6 (2005) 1265–1295.
[30] D.R. Liu, Y.Y. Shih, Integrating AHP and data mining for product recommendation based on customer lifetime value, Information and Management 42 (2005) 340–387.
[31] L. Akoglu, C. Faloutsos, Valuepick: Towards a value-oriented dual-goal recommender system, ICDM Workshop: Optimization Based Techniques for Emerging Data Mining Problems, Sydney, Australia, 2010.
[32] Y. Ge, H. Xiong, A. Tuzhilin, Q. Liu, Cost-aware collaborative filtering for travel tour recommendations. ACM Transactions on Information Systems, 32, 1 (2014), 4.
[33] K. Hosanagar, R. Krishan, L. Ma, Recommended for you: The impact of profit incentives on the relevance of online recommendations, Proceedings of the International Conference on Information Systems, France, 2008.
[34] S. Bag, S.K. Kumar, M.K. Tiwari, An efficient recommendation generation using relevant Jaccard similarity, Information Sciences 483 (2019) 53–64.
[35] S. Bag, A. Ghadge, S.K. Kumar, An integrated recommender system for improved accuracy and aggregate diversity, Computers & Industrial Engineering 130 (2019) 187–197.
[36] T.C. Huang, Y.L. Chen, M.C. Chen, A novel recommendation model with Google similarity, Decision Support Systems 89 (2016) 17–27.
[37] J. Lu, D. Wu, M. Mao, W. Wang, G. Zhang, Recommender system application developments: a survey, Decision Support Systems 74 (2015) 12–32.
[38] H.N. Kim, I. Ha, K.S. Lee, J.S. Jo, A. El-Saddik, Collaborative user modeling for enhanced content filtering in recommender systems, Decision Support Systems 51 (4) (2011) 772–781.
[39] J.L. Herlocker, J.A. Konstan, J. Riedl, Explaining collaborative filtering recommendations, Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, 2000, pp. 241–250.
[40] G. Zacharia, A. Moukas, P. Maes, Collaborative reputation mechanisms in electronic marketplaces, Decision Support Systems 29 (4) (2000) 371–388.
[41] J. Bobadilla, F. Serradilla, A. Hernando, Collaborative filtering adapted to recommender systems of e-learning, Knowledge-Based Systems 22 (2009) 261–265.
[42] S. Ding, Z. Wang, D. Wu, D., D. L. Olson, Utilizing customer satisfaction in ranking prediction for personalized cloud service selection. Decision Support Systems. 93 (2017) 1–10.
[43] A. Jøsang, R. Ismail, C. Boyd, A survey of trust and reputation systems for online service provision, Decision Support Systems 43 (2007) 618–644.
[44] M. Si, Q. Li, Shilling attacks against collaborative recommender systems: a review, Artificial Intelligence Review 42 (4) (2018) 767–799.
[45] S. Prawesh, B. Padmanabhan, The 'Most popular news' recommender: count amplification and manipulation resistance, Information Systems Research 25 (3) (2014) 569–589.
[46] D.K. Chae, S.C. Lee, S.Y. Lee, S.W. Kim, On identifying k-nearest neighbors in neighborhood models for efficient and effective collaborative filtering, Neurocomputing 278 (2018) 134–143.
[47] M. Reusens, W. Lemahieu, B. Baesens, L. Sels, A note on explicit versus implicit information for job recommendation, Decision Support Systems 98 (2017) 26–35.
[48] C.N. Ziegler, J. Golbeck, Investigating interactions of trust and interest similarity, Decision Support Systems 43 (2) (2007) 460–475.
[49] S. Smadi, N. Aslam, L. Zhang, Detection of online phishing email using dynamic evolving neural network based on reinforcement learning, Decision Support Systems 107 (2018) 88–102.
[50] T.W. August, M.F. Niculescu, H. Shin, Cloud implications on software network structure and security risks, Information Systems Research 25 (3) (2014) 489–510.
[51] B. Li, L. Chen, X. Zhu, C. Zhang, Noisy but non-malicious user detection in social recommender systems, World Wide Web 16 (2013) 677–699.
[52] R. Yera, Y. Mota, L. Martínez, Correcting noisy ratings in collaborative recommender systems, Knowledge-Based Systems 76 (2015) 96–108.
[53] E. Anderson, D. Simester, Deceptive Reviews: The Influential Tail. MIT Report Paper, Cambridge, MA (2013).
[54] J.B. Kruskal, Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis, Psychometrika 29 (1964) 1–27.
[55] S. Goode, D. Lacey, Detecting complex account fraud in the enterprise: the role of technical and non-technical controls, Decision Support Systems 50 (2011) 702–714.
[56] C. Posey, T. Roberts, P.B. Lowry, B. Bennett, J. Courtney, Insiders' protection of organizational information assets: development of a systematics-based taxonomy and theory of diversity for protection-motivated behaviors, MIS Quarterly 37 (4) (2013) 1189–1210.
[57] C.N. Ziegler, S.M. McNee, J.A. Konstan, G. Lausen, Improving recommendation lists through topic diversification, Proceedings of the Fourteenth International World Wide Web Conference (2005) 22–32.
[58] B. Sampson, Sell your Book on Amazon: Top-Secret Tips Guaranteed to Increase Sales for Print-on-Demand and Self-Publishing Writers, Outskirts Press, Parker, CO, 2007.
[59] A.V. Bodapati, Recommendation systems with purchase data, Journal of Marketing Research 45 (2008) 77–93.
[60] B. Lika, K. Kolomvatsos, S. Hadjiefthymiades, Facing the cold start problem in recommender systems, Expert Systems with Applications 41 (4) (2014) 2065–2073.

Yuanfeng Cai is an Assistant Professor of Computer Information Systems Department at Zicklin School of Business, Baruch College, City University of New York. Dr. Cai earned her Ph.D. in Information Systems from Iowa State University, and B.E. in Computer Science and Technology from Tongji University, China. Her research interests include

business analytics, data mining and decision-making with imperfect data. She has published papers in INFORMS Journal on Computing, Decision Support Systems, and leading Information Systems conferences including ICIS, AMCIS, HICSS among others.

Dan Zhu is a Professor of Information Systems and Computer Science at Iowa State University. She received her Ph.D. in Information Systems from Carnegie Mellon University. Her research emphasizes the development of computational and analytical models to support business decision makings. Dr. Zhu has published papers in Proceedings of National Academy of Sciences, INFORMS Journal on Computing, Information System Research, Decision Sciences, Naval Research Logistics, Annals of Operations Research, Decision Support Systems, and many other professional journals.