



# A genre trust model for defending shilling attacks in recommender systems

Li Yang<sup>1</sup> · Xinxin Niu<sup>1,2</sup>

Received: 4 August 2020 / Accepted: 27 March 2021  
© The Author(s) 2021

## Abstract

Shilling attacks have been a significant vulnerability of collaborative filtering (CF) recommender systems, and trust in CF recommender algorithms has been proven to be helpful for improving the accuracy of system recommendations. As a few studies have been devoted to trust in this area, we explore the benefits of using trust to resist shilling attacks. Rather than simply using user-generated trust values, we propose the genre trust degree, which differ in terms of the genres of items and take both trust value and user credibility into consideration. This paper introduces different types of shilling attack methods in an attempt to study the impact of users' trust values and behavior features on defending against shilling attacks. Meanwhile, it improves the approach used to calculate user similarities to form a recommendation model based on genre trust degrees. The performance of the genre trust-based recommender system is evaluated on the Ciao dataset. Experimental results demonstrated the superior and comparable genre trust degrees recommended for defending against different types of shilling attacks.

**Keywords** Shilling attack · Collaborative filtering · Recommender system · Trust value · Genre trust

## Introduction

To overcome the information overload problem, recommender systems use specific characteristics of users and items to filter out the content required by users and generate personalized recommendations. Trust is receiving considerable attention in the academic community and e-commerce industry and plays a central role in exchange relationships involving unknown risk [1, 2]. It is observed that trust has a significant correlation with user preference similarities in rating systems [3], especially in the e-commerce industry [4]. Thus, most existing work about trust has been studied in the context of rating prediction [5, 6], although user trust may be violated if a recommender system reduces its accuracy intentionally. Users with similar preferences are more

likely to trust each other, demonstrating homophiles [7] in the trust network. Previous work suggests that a model combining rating similarity with trust networks can achieve better performance than the trust propagation model [8, 9].

However, user's trust relationships usually follow a power-law distribution, and a large, long tail of users only have few trusted or distrusted users [10]. Furthermore, recent state-of-the-art methods have relied on the learning of even larger, more complex factorization models, often using nontrivial combinations of multiple submodels [11, 12].

Thus, in this paper, we improve the calculation algorithm of the trust matrix with the genre trust degree (GTD) and combine GTD with user similarities. The main idea of our model is the synthesis of user credibility and the trust degree between users. We promote the weight of high-credibility users to make better recommendations. In summary, we make the following main contributions: first, we propose a new method to detect shilling attackers with higher accuracy and ensure the stability of the recommendation results. Second, we propose an improve trust degree calculation algorithm to resist shilling attacks.

This paper is structured as follows: In section “[Related work](#)”, we refer to related work in the area and explain the differences between our study and theirs. Section “[Trust degree in recommender system](#)” briefly describes the definition of trust.

✉ Li Yang  
liziy1988@aliyun.com; yangli1988@bupt.edu.cn  
Xinxin Niu  
xxniu@bupt.edu.cn

<sup>1</sup> Computer Science and Technology, Beijing University of Posts and Telecommunications, Beijing, China

<sup>2</sup> Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University, Guiyang, China

Then, we provide a detailed description of the GTD model in section “Genre trust degree model for defending shilling attacks”. We present our real data-based experiment results in section “Experimental setup”. Finally in section “Results analysis”, we conclude this paper and present our future research directions.

## Related work

Recommender systems have taken on an impressive role in our daily life, to the extent that profile-injection attacks directed at misleading recommendation results appear continuously. The shilling attack [13] is one type of the profile-injection attacks that is regularly used, and research in the area of shilling attacks has made significant advances in the last few years. Shilling attacks aim to control item ratings for their own purposes. From the intention perspective, shilling attacks were classified into two basic types in researchers’ early work: the push attack and “nuke” attack. The general form of a shilling attack is depicted in Table 1. “Item” refers to items evaluated by users in the system, such as movies, books, merchandise, and so on.

A shilling attack profile generally consists of ratings of selected items, filler items, unrated items, and the target item. Various attack models have thus been discovered and appropriate metrics developed to measure the effectiveness of an attack [14]. Different shilling attack models correspond to different methods of choosing selected items and filler items. The commonly used methods of shilling attacks are the random attack, average attack, bandwagon attack, segmented attack, and sampling attack: (a) Random attack: The ratings of filler items are chosen randomly and the target item is assigned with a pre-specified rating. The selected item sets are empty. (b) Average attack: A new attack model with a better attack effect than random attack. The selected items sets are still empty, while the ratings of filler items are the individual mean for that item rather than the global mean. (c) Bandwagon attack: Attackers pick up a part of frequently rated items in system as the selected items. The filler items are randomly chosen while the rating value for them is the mean across the whole system. (d) Segmented attack: Avoiding being detected, attackers give high ratings to those items similar to target items. (e) Sampling attack: This attack model needs to know more system information for copying existing user models. Attackers imitate real users selecting and rating items, while the target item are assigned with the highest rating.

**Table 1** General form of shilling attack profile

	Selected	Filler	Unrated	Target
Items	$I_l \sim I_k$	$I_{k+l} \sim I_m$	$I_{m+l} \sim I_n$	$I_{n+l} \sim I_r$
Ratings	$R_l \sim R_k$	$R_{k+l} \sim R_m$	$R_{m+l} \sim R_n(\text{null})$	$R_{n+l} \sim R_r$

The detection of shilling attacks on recommender systems has been studied by many researchers. The use of clustering methods to filter out shilling attackers or the analysis of normal users’ behavior patterns to detect shilling attackers has been studied by many researchers. Alostad et al. [15] presented an improved support vector machine (SVM) and Gaussian mixture model (GMM)-based shilling attack detection algorithm (SVM-GMM). Samaiya et al. [16] combined PCA and SVM—two classification algorithms—to improve the effect of the defense against shilling attacks. Lee et al. [17] proved that users’ evaluations are affected by other users and that users connected by a network of trust exhibit significantly higher similarity on items. Ardisson et al. [18] proposed a compositional recommender system based on multi-faceted trust using social links and global feedback about users. Jaehoon et al. [19] proposed a new trust recommendation algorithm, TCRec. Pan et al. [20] presented an adaptive learning method to map trust values to the [0, 1] interval and use “directionality” to distinguish users.

However, the limitation that these methods need more extraneous information is also obvious, such as user type and item category, and they still suffer from the problem of cold starting. Compared with the existing methods, the model proposed in this work can result in more efficient recommendations using limited existing information. Additionally, the credibility of users is a kind of real user interaction data that can be referred to, and we integrate it into the system model not only to better eliminate the interference of attacking users, but also can ameliorate the restriction of data sparsity. At the same time, the calculated recommendation results are more in line with user preferences to a certain extent.

## Trust degree in recommender system

In this section, we will first give the definition and properties of trust in recommender systems. Then, we recommend several often-mentioned trust calculation models.

### Definition of trust

The formation of a trust relationship requires the interaction of real users. Because of the lack of real interaction behaviors, trusted attack users are easily overlooked in the trust relationship. Therefore, the impact of the trusted attacking user on the normal recommendation system is reduced.

Trust is generally considered to be a complex concept; two common definitions of trust, *reliability trust* [21] and *decision trust* [22], are most widely used. However, in recommender systems, trust is mostly defined as being correlated with similar preferences toward items commonly rated by two users [23, 24]. Guo et al. [25] presented a more

unambiguous definition: trust is defined as one's belief in the ability of others to provide valuable ratings.

With intensive and deeper study, trust measurement methods can be divided into global trust and local trust [26]. It has been found that local trust methods perform better in terms of resisting shilling attacks [27]; furthermore, researchers have summarized some distinct properties of trust, described as follows [28, 29]:

- **Asymmetry/subjectivity:** A user may hold different opinions toward different target users, and different users also may have various opinions or trust degrees with the same user. Thus, we can have  $trust(u, v) \neq trust(v, u)$ .
- **Transitivity:** If user  $u$  trusts user  $v$ , and user  $v$  trusts user  $w$ , we believe that user  $u$  trusts user  $w$  to same degree.
- **Dynamicity/temporality:** Trust is the users' previous interactions and it changes over time.
- **Context Dependence:** Trust is context-specific, which indicates a user who is trustable in art may not be helpful in computer science.

## Trust calculation methods

Different trust calculation models use different trust metrics to calculate user trust from user ratings, and most of these methods are based on the assumptions that users with similar ratings tend to be trustworthy. Researchers prefer to regard user similarity as inferred trust values. One representative method was proposed by Papagelis et al. [30], who take the Pearson correlation coefficient as the calculation method of trust, as follows:

$$s_{u,v} = \frac{\sum_i (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_i (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_i (r_{v,i} - \bar{r}_v)^2}}, \quad (1)$$

where  $s_{u,v}$  is the similarity of  $user_u$  and  $user_v$ , and the trust value is assigned as similarity. Some researchers have set a certain similarity as the threshold to filter trustworthy users. Although one characteristic of similarity trust calculation method is symmetry, Sotos et al. [31] proposed that the Pearson correlation coefficient is not transitive unless users are highly correlated. Hwang and Chen [32] obtained the trust score by averaging the prediction error on co-rated items, as follows:

$$t_{u,v} = \frac{1}{|I_{u,v}|} \sum_{i \in I_{u,v}} \left( 1 - \frac{|\bar{r}_u + (r_{v,i} - \bar{r}_v) - r_{u,i}|}{r_{\max}} \right). \quad (2)$$

However, the current trust metrics are not satisfactory for producing distinguishable trust lists and may be further limited by the used similarity measures or required thresholds.

Moreover, the existing trust metrics are generally based on symmetric methods, such as similarity and error measures; thus, they all can be treated as similarity-based trust metrics. The model described in this paper improves trust calculation methods by distinguishing trust value by ratings in different genres and providing a more reasonable user similarity calculation algorithm.

## Genre trust degree model for defending shilling attacks

In this section, we introduce our method, the GTD model, to resist shilling attacks, including our improved recommendation method. After introducing the genre trust degree, we present our proposed model with its optimization method. Prior to delving into our GTD model in recommender systems, we define the notation used in our paper. We denote  $U, I, R$ , and  $K$  as the set of all users, items' ratings, and item-genres in recommender systems, respectively. For simplicity, suppose our experimental datasets have an  $m \times n$  rating matrix, and we keep symbols  $u, v$  for users and  $i, j$  for items; thus,  $r_{u,i}$  represents a rating given by user  $u$  on item  $i$ , and  $I_u$  is the set of items rated by user  $u$ ,  $R = \{r_{u,i}\}$  denoting  $m$  users' ratings on  $k$  genre items. As users record their trust relationships in the trust lists, we let  $t_{u,v}$  be the trustworthiness of user  $v$  toward user  $u$ , and  $T = \{t_{u,v}\}$  presents the trust list network.

## Genre trust degree

Many *trust metrics* have been proposed to calculate implicit trust from user ratings, mainly based on the intuition that users whose ratings are close or similar to each other tend to be trustworthy [33]. In our paper, we propose that trust should be useful not only to generate item predictions but also to suggest reliable users. In contrast to traditional user trust calculation methods, there are two parts to the *trust degree* in the GTD model: the traditional trust value between users and the credibility of users. In most proposed trust degree computing methods, if  $user_u$  trusts  $user_v$  and  $user_w$ , when giving recommendations to  $user_u$ , the influence of  $user_v$  is identical with  $user_w$ , because they have the same trust degree. The trust theory indicates that "context dependence" is one main character of trust. It shows that users have different trust degrees in different fields, whereas the traditional trust degree computing methods leave this out of consideration. In this paper, we suppose that users have their own preferences, which leads to users better understanding their familiar areas. These scores provide a better reference for providing recommendations to other new users.

As users are more indicative in their familiar or favorite areas, we introduce "user credibility" in our model to

distinguish different trustee reliabilities in different areas. When calculating users' trust degrees, instead of giving a single-fixed value, we propose corresponding trust degrees according to the genres of items. First, we briefly describe some essential notations used in our method. We use the review ratings to calculate user credibility, and  $\text{reRating}_{v,s,r}$  stands for the review rating user<sub>s</sub> gives to the review<sub>r</sub> made by user<sub>v</sub>.  $\text{Gen}(r)$  is the genre of the item to which review<sub>r</sub> refers. The sum of the review ratings of the reviews user<sub>v</sub> provides for those items belonging to genre  $k$  is given by:

$$\text{ReRating}(v, k) = \sum_{S, \text{Gen}(r)=k} \text{reRating}_{v,s,r}. \quad (3)$$

Based on Eq. (3), we can obtain the credibility of user<sub>v</sub> for the items of genre  $k$ , where  $N$  is the total number of item-genres:

$$\text{Credibility}_{v,k} = \frac{\text{ReRating}(v, k)}{\sum_k \text{ReRating}(v, k)} \times N. \quad (4)$$

Thus, the genre trust degree of trustor<sub>u</sub> to trustee<sub>v</sub> on genre  $k$  is as follows:

$$\begin{aligned} \text{GenreTrust}_{u,v,k} &= \text{TrustValue}_{u,v} \times \text{Credibility}_{v,k} \\ &= \text{TrustValue}_{u,v} \frac{\sum_{S, \text{Gen}(r)=k} \text{reRating}_{v,s,r}}{\sum_k \text{ReRating}(v, k)} N, \end{aligned} \quad (5)$$

where  $\text{TrustValue}_{u,v}$  is the total trust value between trustor<sub>u</sub> and trustee<sub>v</sub>, which is usually a single-fixed value and is defined as follows:

$$\text{TrustValue}_{u,v} = \begin{cases} 1, & \text{trust} \\ \frac{\sum_i (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_i (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_i (r_{v,i} - \bar{r}_v)^2}}, & \text{unknown} \\ 0, & \text{distrust} \end{cases}. \quad (6)$$

We combine the traditional user trust value in Eq. (1) with user credibility to determine the genre trust degree, and this fully demonstrates the “context dependence” characteristic of trust.

## Genre trust model

As we know, users are more inclined to trust those users who have similar interests or behavior patterns. Therefore, some researchers introduced the notion of trust to collaborative filtering (CF) recommender systems to improve the system's robustness to malicious attacks, especially shilling attacks. Because the shilling attacker is a fake user, it can only disguise its own user behavior, but lacks interaction data with real users. Therefore, combined with the CF algorithm can well resist the interference of shilling attacks.

CF recommendation algorithms are generally divided into two types: item-based and user-based collaborative filtering recommendation methods. The common steps of CF algorithms' recommendations to users are shown in Fig. 1:

Based on traditional CF recommendation algorithms, the GTD model uses the genre trust degree to improve the user similarity calculation algorithm. The common steps of our GTD model are shown in Fig. 2:

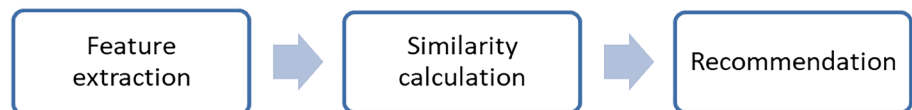
And the algorithm flowchart of GTD model is shown in Fig. 3:

1. According to Eq. (3), we calculate the genre-trusts between each user for items belonging to different genres:

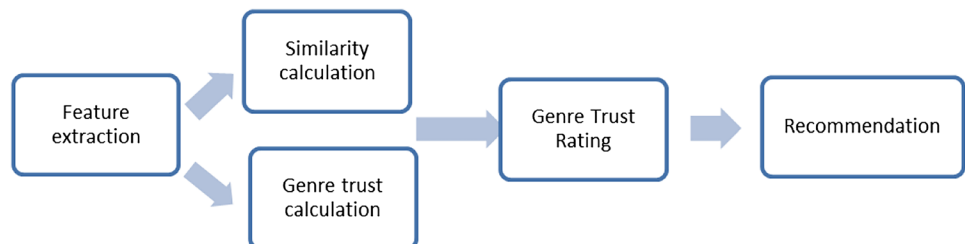
$$GT = \{\text{GenreTrust}_{u,v,k}, u, v \in U, k \in K\}, \quad GT \in R^{m \times m \times k}. \quad (7)$$

2. We calculate all similarities between users and then sort each user's *user-similarity* set,  $\text{Sim}_u = \{\text{Similarity}_{u,v}, v \in U\}$ , in descending order and select the top  $s$  similar users as the corresponding user's *similar-user* set; that is,  $\text{SimUser}_u$ .

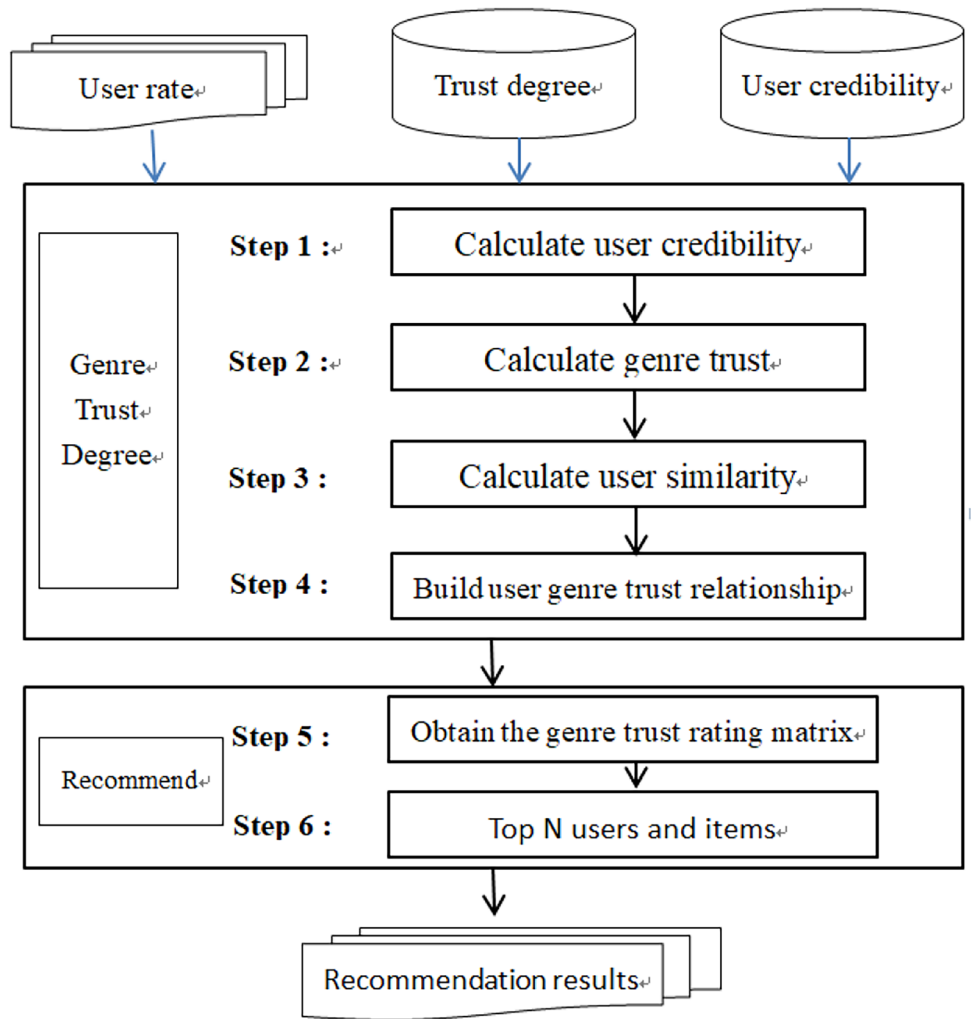
**Fig. 1** Steps of collaborative filtering (CF) recommendation algorithms



**Fig. 2** Steps of the genre trust degree (GTD) recommendation model



**Fig. 3** Algorithm flowchart of the genre trust degree (GTD) recommendation model



There are many ways to calculate user similarity. In the experimental part of this article, we choose Cosine correlation coefficient, Pearson correlation coefficient, and trust correlation coefficient for user similarity calculation methods for comparison experiments.

3. We take the ratings of these items rated by user<sub>v</sub> in the set SimUser<sub>u</sub> as similar users' rating lists to user<sub>u</sub>:

$$SR = \{SimRating_u\} = \{r_{v,i}, v \in SimUser_u\}. \quad (8)$$

4. Then, we use the corresponding genre-trusts of user<sub>u</sub> to weight these ratings to obtain the genre trust rating matrix GR, and this is given by:

$$GenreTrustRating_{u,i} = GenreTrust_{u,v,k} \times SimRating_{v,i},$$

where  $Gen(i) = k$ .

$$(9)$$

5. For item<sub>i</sub>, we add up each item's genre trust rating as follows, sort all items' genre trust ratings in descending order, and select the top  $r$  items as the final items we recommend to corresponding user<sub>u</sub>:

$$GR_i = \sum_{u \in SimUser_u} GR_{u,i}. \quad (10)$$

Thus, the predicted score of user<sub>u</sub> for the system item  $i$  can be calculated as follows:

$$\begin{aligned} p_{u,i} &= \sum_{v \in SimUser_u, Gen(i)=k} GenreTrustRating_{v,i} \\ &= \sum_{v \in SimUser_u} TrustValue_{u,v} \frac{\sum_{S, Gen(r)=k} reRating_{v,s,r}}{\sum_k ReRating(v, k)} N \\ &= \sum_{v \in SimUser_u} \frac{\sum_i (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_i (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_i (r_{v,i} - \bar{r}_v)^2}} \frac{\sum_{S, Gen(r)=k} reRating_{v,s,r}}{\sum_k ReRating(v, k)} N. \end{aligned} \quad (11)$$

The GTD model combines the user trust value with user credibility, providing a better recommendation method to



resist different types of shilling attacks and recommend items to users. Our GTD model outperforms state-of-the-art algorithms in the following aspects: (1) combining user credibility with trust value is more reasonable and comprehensive than only considering the one-way trust value given by users; (2) the trust degree can compensate for the weaknesses of recommender system under shilling attacks; (3) the method weight items ratings with trust degrees—that is, combining the user-based recommendation algorithm with an item-based recommendation algorithm—to increase the results' diversity and give our system more robustness.

## Experimental setup

In this section, we first introduce the detail about the Ciao dataset used in our experiments, and then describe our experimental procedures and the results of applying our GTD model to recommend items to users in these datasets under shilling attacks.

### Dataset analysis

The Ciao dataset presents the trust links directly in a trust file as trustor, trustee, and trust value, with 17,615 users and 16,121 movies. In the Ciao dataset, users do not only rate movies, but give reviews to movies. Furthermore, users can provide ratings for these movie reviews even if they have not provided ratings for these movies. There are 72,665 movie reviews and 1,625,480 review ratings. The dataset consists of three files, which are *movie-rating.txt*, *review-rating.txt*, and *trusts.txt*, and the specific data contents contained in each file are shown in Table 2. And the specific statistics of Ciao dataset shows that more than 76% ratings are high scores, while the data in Ciao dataset are very spares.

**Table 2** Ciao dataset description

File	File name	Description
Movie rating	userID	Users who make ratings
	movieID	Scored movie
	genreID	Movie category
	reviewID	Users who give reviews
	movieRating	User's ratings of movies
	date	Date
Review rating	userID	The user who evaluates the evaluate
	reviewID	Users who give reviews
	reviewRating	User's ratings of the review
Trusts	trustorID	Users who make trust evaluations
	trusteeID	Trusted users
	trustValue	0 means distrust, 1 means trust

### Generate attack data

We divide the Ciao dataset into two parts: training and test parts, and in our experiments, tenfold cross validation is used. For each experiment, we take ninefold as the training set and the last one as the test set. Ten rounds of executions are conducted, and then, we adopt the average result of them as the final performance result in one set of parameters. However, the extent of the data sparseness in the Ciao dataset is too severe; if we experiment directly on the original dataset, the performances of shilling attacks and recommendation algorithms will be too slight to reflect the improvement of our GTD model. For this reason, we preprocess the original dataset to select those users who rated at least 20 items with items that were rated at least 10 times.

As the aim of a shilling attack is to push or “nuke” the target items' ratings and increase or decrease the recommendation times of target items, we take the random attack, average attack, and bandwagon attack—three frequently used shilling attack models—as the attack data generation methods in our experiment: the random attack and average attack are easy to implement, while the bandwagon attack has better attack effects [34]. We inject these attack data into the system to simulate shilling attack process, and due to newly attacked users often lacking trust relationship data, we randomly select trustees and choose the average trustee number in the original system as the attackers' trustee number. In contrast to ordinary shilling attack data, we need to generate the attackers' trust relationship. As with the pattern to determine the attackers' profile size, we take the average number of users in the system user trust list as our attacker trust list size. Thus, to simulate real users and generate better attack effects, we take the attackers' trustors from the users with a high trust value. Then, we introduce some dimensions of the attack data:

- **Attack size:** Attack size is the number of attackers. Only a few malicious users can mislead recommender systems to make inappropriate judgments, even though they do not know the implementation detail or algorithm of the systems [35].
- **Profile size:** Profile size is the rating number of a given attack profile. To better simulate real users' rating patterns of behavior, we use the system average rating number as our shilling attackers' profile size.

We set the attack sizes at 5%, 6%, 7%, 8%, 9%, and 10%, and the profile size is 20 in the three types of shilling attack methods simulated in our experiment. We conduct a series of contrast experiments using the cosine and Pearson correlation coefficient collaborative recommendation models, and they are defined as follows:

- **Cosine correlation coefficient:**

$$\text{Cos}_{u,v} = \frac{\sum r_{u,i} r_{v,i}}{\sqrt{\sum r_{u,i}^2 \sum r_{v,i}^2}}. \quad (12)$$

- Pearson correlation coefficient:

$$\begin{aligned} \rho_{u,v} &= \frac{\text{cov}(u,v)}{\sigma_u \sigma_v} = \frac{E((u - \mu_X)(v - \mu_Y))}{\sigma_u \sigma_v} \\ &= \frac{\sum (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum (r_{u,i} - \bar{r}_u)^2 \sum (r_{v,i} - \bar{r}_v)^2}}. \end{aligned} \quad (13)$$

Meanwhile, we also compare the recommendation results using the cosine and Pearson correlation coefficient as user similarity calculation methods in step 2 in chapter 4.2 of the GTD model to test and verify the robustness of our model.

## Performance measures

- MAE

*MAE*, short for mean absolute error, calculates the average absolute difference between the predicted ratings and real user ratings for all users and items in the dataset before and after a specified pattern shilling attack. *MAE* is defined as follows:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |f_i - y_i| = \frac{1}{n} \sum_{i=1}^n |e_i|, \quad (14)$$

where  $n$  is the number of predicted item ratings. For a specified item  $i$ ,  $y_i$  is the real user rating, while  $f_i$  is the predicted user rating of item  $i$ . In our experiments, item  $i$  is the target item ID of shilling attackers.

- RecRate

Shilling attacks usually aim to deliberately increase or decrease recommendations for items. Thus, if shilling attack data are injected into the recommender system, the target items' recommendations must be increased or decreased intentionally. The target item recommendation rate for one recommendation under a shilling attack is defined as follows:

$$\text{recRate}_{\text{target}} = \frac{\sum_u \text{Recommend}_{u,\text{target}}}{N}, \quad (15)$$

where  $u$  is the user in the test dataset and  $\text{Recommend}_{u,\text{target}}$  equals 1 if the target item has been recommended to  $user_u$ ; otherwise, it equals 0.  $N$  is the number of users in the relative test dataset.

## Results analysis

To verify the performance of the GTD model, first, to evaluate the improvement results of the GTD model in defending against shilling attacks compared with traditional CF recommendation algorithms and the EnsembleTrustCF algorithm, we use Eq. (12) to present the target item recommendation rate to users under different types of shilling attacks and take the original recommendation rate as baselines. Figure 4 shows the recommended rate (recRate) of target items using the recommendation methods of the traditional CF recommendation algorithm using correlation coefficient similarity computing methods and the GTD model using correlation coefficient similarity computing methods with no shilling attack data and we take them as the baselines. The movieID of the target item is "300". Figures 4, 5 and 6 present the recRate of the target item under different shilling attack sizes, which are 5%, 6%, 7%, 8%, 9%, and 10%.

## Benchmark recommendation ratio

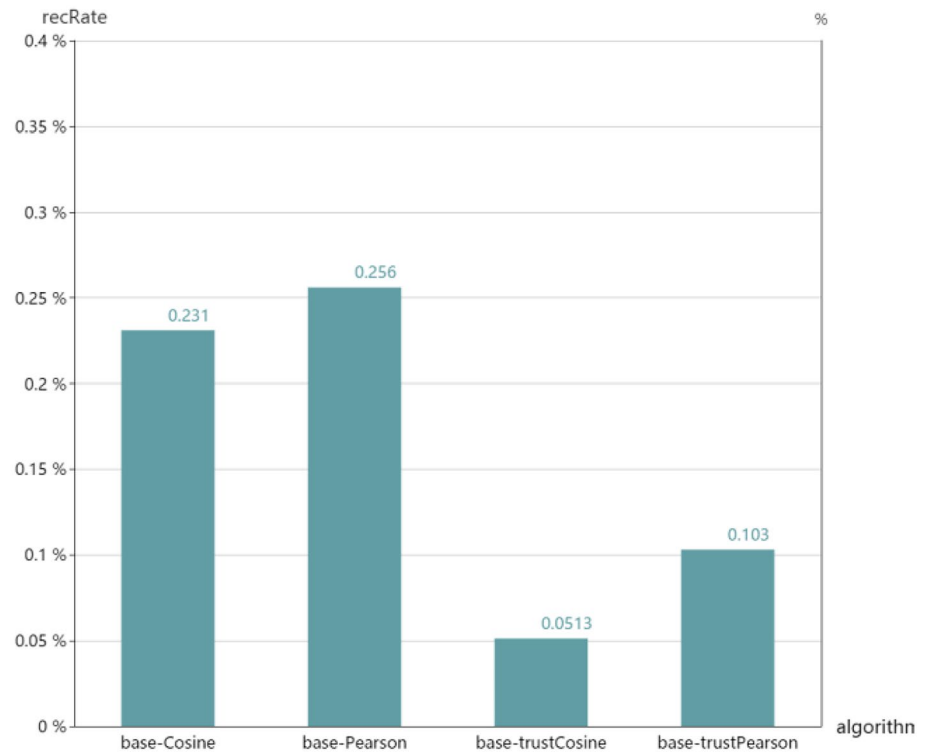
Figure 4 shows that for both traditional collaborative filtering and GTD recommendation methods, the recRate of the target movie with the cosine correlation coefficient is higher than the Pearson correlation coefficient. In the meantime, the GTD model has a lower recRate than traditional collaborative filtering models generally, and this also shows that the GTD model has an impact on the diversity of item recommendations. In our experiments, we set these four data values as baselines and compare them with the recRate of the target movie after injecting shilling data into the system. The comparison results reflect different recommendation models defending against the effects of shilling attacks.

## Experimental results and analysis

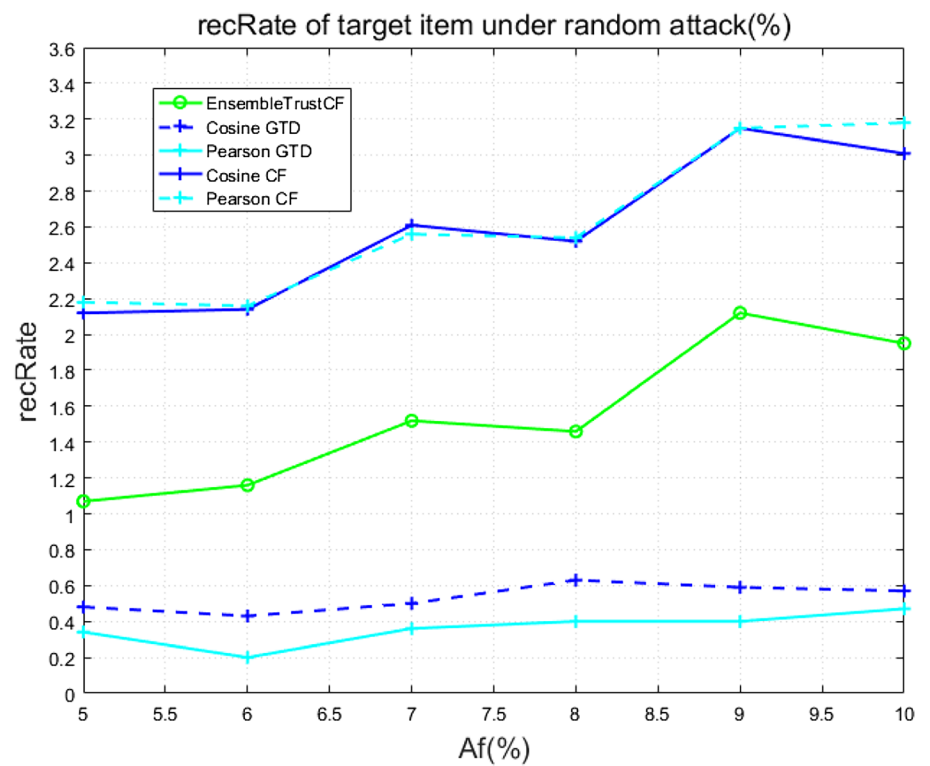
To more comprehensively verify the validity of the GTD model, the EnsembleTrustCF algorithm is selected for the comparison experiment. The EnsembleTrustCF algorithm is a recommendation algorithm based on a trust network proposed by Victor [36] et al. This algorithm takes trust value as the main weight consideration when predicting user ratings and has high coverage rate and prediction accuracy. The formula for the EnsembleTrustCF algorithm to calculate user-item ratings is as follows:

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u \in R^T} t_{a,u} (r_{u,i} - \bar{r}_u) + \sum_{u \in R^S \setminus R^T} s_{a,u} (r_{u,i} - \bar{r}_u)}{\sum_{u \in R^T} t_{a,u} + \sum_{u \in R^S \setminus R^T} s_{a,u}}, \quad (16)$$

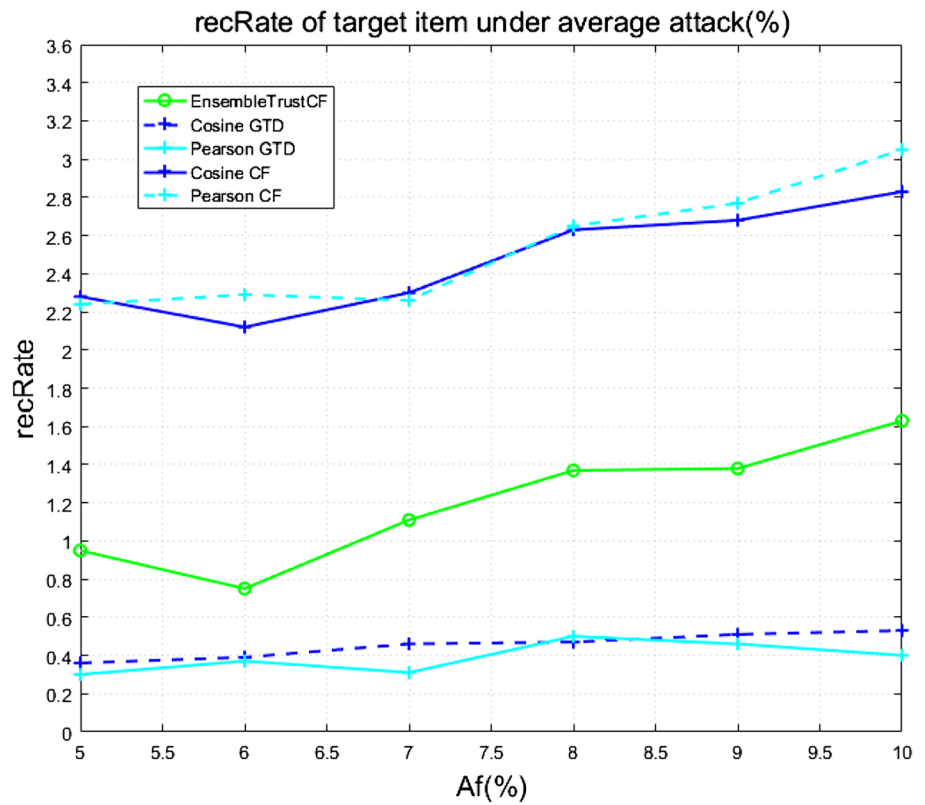
**Fig. 4** recRate of CF recommendation algorithms and GTD model with original data



**Fig. 5** Recommendation rate (recRate) under random attack





**Fig. 6** recRate under average attack

where  $P_{a,i}$  is the predicted rating that user<sub>*a*</sub> gives to item<sub>*i*</sub>,  $r_a$  is the average rating of user<sub>*a*</sub> and  $R^T$  indicates the set of users trusted by user<sub>*a*</sub>.  $R^S$  indicates a similar-user set to user<sub>*a*</sub>.  $S_{a,u}$  is the rating similarity between user<sub>*a*</sub> and user<sub>*u*</sub>.  $t_{a,u}$  is the trust value of user<sub>*a*</sub> to user<sub>*u*</sub>. From the existing experimental results, it can be seen that the defense effect of using the Pearson correlation coefficient to calculate user similarity is better than that of using the cosine correlation coefficient. Therefore, we also use the Pearson correlation coefficient to calculate user similarity in the EnsembleTrustCF algorithm.

In this part, the experimental results of five algorithms under three different shilling attack methods are compared. They are the CF recommendation algorithm using the cosine correlation coefficient and using Pearson correlation coefficient, the GTD recommendation model using the cosine

correlation coefficient and using Pearson correlation coefficient, and the EnsembleTrustCF algorithm. The detailed data are presented in Tables 5, 6, 7 and 8.

### Effectiveness

We split the Ciao dataset into the training set and test set randomly. Then, we used different methods to calculate user similarities, create recommendation lists, and obtain statistics for the accuracy of these algorithms to prove the validity of the GTD model. The accuracy of the recommendation results of the Ciao dataset without shilling attack data with different test set sizes is shown in Table 3. Although different training sets may influence the results, the overall trend is

**Table 3** Accuracy without shilling attack data

Test dataset size (%)	Cosine CF	Pearson CF	Pearson EnsembleTrustCF	Cosine GTD	Pearson GTD
5	0.9065	0.9463	0.9574	0.9502	0.9615
10	0.9189	0.9537	0.9593	0.9546	0.9641
15	0.9217	0.9520	0.9618	0.9554	0.9648
20	0.9018	0.9491	0.9633	0.9567	0.9664

obvious: the GTD model performs better than the traditional CF recommendation algorithms and EnsembleTrustCF. The Pearson correlation coefficient has higher accuracy than the cosine correlation coefficient.

## recRate

In this part, we compare the recRate among the GTD model, traditional CF recommendation algorithms, and EnsembleTrustCF on the Ciao dataset. Defending against shilling attacks means that the recommendation results should not have an obvious change after injecting attack data. Thus, a lower recRate represents better performance.

### 1. recRate under Random Attack

Figure 5 shows that the traditional CF recommendation algorithms, regardless of using the cosine or Pearson correlation coefficient similarity computing methods, are sensitive to random attacks: the rates of target items to be recommended have significant overall increases even if the attack size is small. On the other hand, the rates do not increase obviously when we use the GTD model. Furthermore, the EnsembleTrustCF algorithm is better than the GTD model with the cosine correlation coefficient, while the performance is worse than the GTD model with the Pearson correlation coefficient.

Moreover, despite the possible presence of some distinctions, the overall trend is that a larger shilling attack size leads to better performance in misleading recommender systems to recommend the target item more frequently. Under the same shilling attack circumstances, the Pearson correlation coefficient CF recommendation algorithm yields better results for defending against shilling attacks than the cosine correlation coefficient method. The results of the GTD model show that the Pearson correlation coefficient performs better than the cosine method.

### 2. recRate under Average Attack

Figure 6 shows similar experimental results to Fig. 5. Compared with Fig. 5, we can conclude that the recommendation effectiveness using cosine or Pearson correlation coefficient CF algorithms is similar under random attack and average attack. Besides, the target item recommended rates using the GTD model are slightly higher above the baselines than without shilling attack data. For shilling attack methods with random factors, the choice of the method for calculating user similarity is also one of the important factors affecting the algorithm effect. In the meantime, the shilling attack effect

does not increase in proportion to the shilling attack size, meaning that only a few malicious users can mislead recommender systems to give inappropriate results. However, when using the same similarity calculation method, the GTD model which we proposed can achieve better shilling attack defense effects.

### 3. recRate under Bandwagon Attack

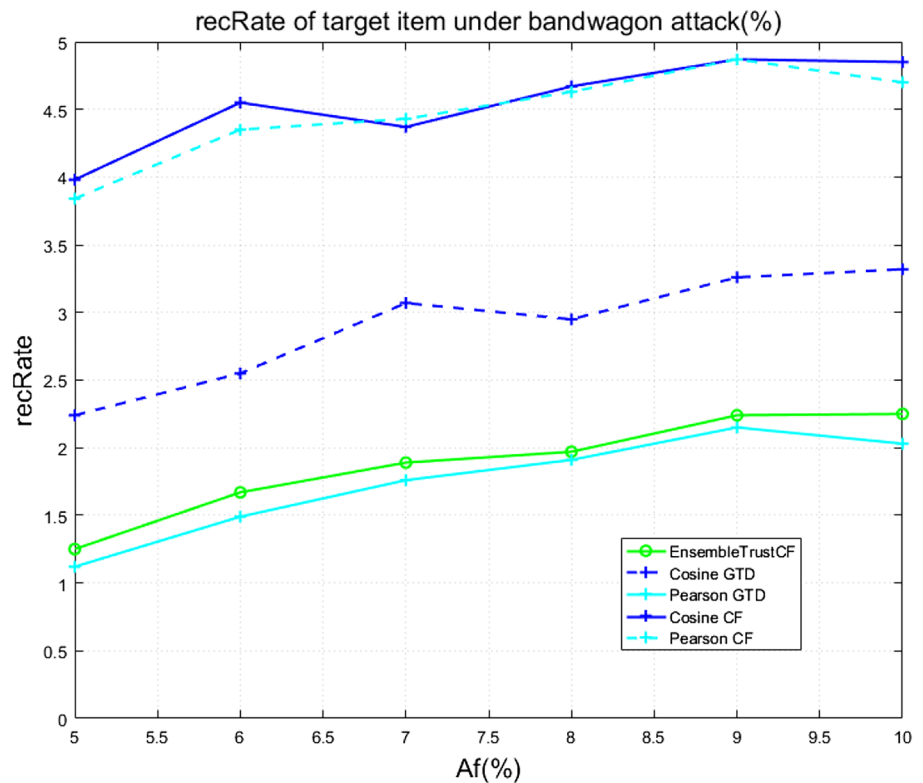
Figure 7 presents the recommendation rates of the target item under bandwagon attack. It is observed that the bandwagon attack has the best attack effect compared to random attack and average attack: the recommendation rates of the target item are improved to varying degrees. It can be seen that the recommendation rate of target items has been improved to different degrees after using bandwagon attacks. However, in contrast to random attacks and average attacks, the GTD recommendation algorithm has a better defense effect than the EnsembleTrustCF algorithm when using either the cosine correlation coefficient or Pearson correlation coefficient. Moreover, when the attack size increases, the recommended rate of shilling attack targets to normal users can be well controlled in the GTD model.

## Mean average error

We select an attack size varying from 5 to 10% for the bandwagon attack and calculate the MEA over the test set: we choose 20% user profile data from the Ciao dataset as the test set. Then, we compare the MAE results under the four situations: cosine CF model, Pearson CF model, cosine GTD model, and Pearson GTD model. The results are shown in Table 4.

After interpreting the data within our experiments, we can come to the following conclusions:

1. The bandwagon attack has the best shilling attack effect, while the random attack is similar to average attack;
2. CF recommender systems are susceptible to shilling attacks, and only a small amount of malicious data can lead to visible attack effects;
3. The GTD model has very acceptable efficiency in defending against a shilling attack and recommending items to users;
4. The Pearson correlation coefficient is better than the cosine correlation coefficient in the GTD model and traditional CF recommender algorithms;
5. In some cases, the customized filtering conditions in the GTD model are so strictly restricted that they may impose limitations in terms of recommendation genres and selection range. This is an area which requires improvement in our future research.

**Fig. 7** recRate under bandwagon attack**Table 4** Overall mean average error (MAE) of bandwagon attack

Attack size (%)	Cosine CF	Pearson CF	Cosine GTD	Pearson GTD
5	0.8091	0.7304	0.6781	0.6670
6	0.8076	0.7237	0.6793	0.6672
7	0.8097	0.7321	0.6809	0.6714
8	0.8110	0.7342	0.6822	0.6732
9	0.8119	0.7335	0.6846	0.6773
10	0.8126	0.7348	0.6871	0.6802

## Conclusions

Based on the intuition that, in recommender systems, users with higher credibility have more weight in recommendations and a user's credibility differs for genres of items, we proposed the GTD model for trust-aware recommendation by providing genre trust degrees and improving the user similarity calculation algorithm. Compared with the traditional trust metric models, our GTD model focuses on high-credibility users and it can better handle new users'

recommendation problems. Moreover, the GTD model raises the cost of shilling attack injections; high-credibility users are needed in advance to interfere with system recommendation results, and this can support effective and sensible recommendations. The experimental results and extensive comparisons discussed in this paper have shown that our GTD model achieves better or comparable performance in defending different attack sizes of shilling attacks than the other two representative trust metrics. The experimental results of the recRate clearly show that the GTD model can control the rate of target items being recommended to users improperly within a certain range and ensures the reliability of recommendation systems.

Since user ratings are time-sensitive, in further study, we will include time sensitivity and ameliorate the issue of lacking new items in recommendations to adapt to the time-sensitive requirements of the recommendation system.

## Appendix

See Tables 5, 6, 7 and 8.

**Table 5** recRate of CF recommendation algorithm using Cosine correlation coefficient

	%	Test1	Test2	Test3	Test4	Test5	Test6	Test7	Test8	Test9	Test10
5%	Random	2.83	1.82	2.42	1.62	2.42	2.22	1.82	2.22	1.41	2.42
	Average	1.82	3.03	2.42	2.63	2.02	3.03	1.82	1.82	2.22	2.02
	Bandwagon	3.64	4.65	4.24	3.23	3.84	3.84	3.23	5.45	4.04	3.64
6%	Random	1.76	1.57	1.76	2.75	1.57	2.75	1.96	2.94	2.16	2.16
	Average	2.55	1.96	1.96	1.96	3.33	2.16	1.76	2.394	1.57	0.98
	Bandwagon	4.31	3.53	4.31	4.51	4.71	5.10	4.90	4.71	5.10	4.31
7%	Random	2.59	2.59	3.33	2.78	2.22	2.04	2.59	2.96	2.78	2.22
	Average	1.67	2.22	2.22	1.67	2.04	2.78	1.85	2.96	2.59	2.96
	Bandwagon	4.81	4.44	4.44	4.26	4.81	4.81	4.81	3.89	3.89	3.52
8%	Random	2.70	2.16	3.24	1.80	2.70	2.52	1.62	2.88	3.42	2.16
	Average	2.16	3.24	2.16	1.98	2.88	2.88	2.52	2.16	2.88	3.42
	Bandwagon	4.68	4.50	4.86	4.68	5.05	4.32	4.50	4.50	4.68	4.86
9%	Random	3.25	3.42	2.74	2.22	3.42	3.59	3.25	3.08	3.25	3.25
	Average	2.91	1.88	2.56	2.91	2.74	2.74	2.91	2.39	2.91	2.91
	Bandwagon	5.64	4.44	4.44	4.79	5.13	5.13	4.96	4.76	5.13	4.27
10%	Random	3.00	3.33	3.17	3.17	3.17	2.83	2.83	2.83	3.50	2.67
	Average	2.83	2.17	3.00	3.00	2.67	3.00	2.67	3.00	3.17	2.83
	Bandwagon	4.67	4.50	5.00	5.33	4.83	4.83	4.50	5.17	4.33	5.33

**Table 6** recRate of CF recommendation algorithm using Pearson correlation coefficient

	%	Test1	Test2	Test3	Test4	Test5	Test6	Test7	Test8	Test9	Test10
5%	Random	2.02	1.62	2.42	2.02	3.23	1.82	2.02	2.63	1.82	2.22
	Average	2.22	2.83	2.42	2.83	1.82	2.42	2.02	1.82	2.02	2.02
	Bandwagon	3.03	4.44	4.44	3.03	3.84	3.84	3.84	4.44	3.84	3.64
6%	Random	1.96	1.18	1.96	2.55	1.57	2.16	1.76	3.14	2.55	2.75
	Average	2.75	2.35	2.35	1.76	2.94	2.16	1.76	2.94	2.16	1.76
	Bandwagon	4.12	3.53	3.92	4.71	4.51	5.29	4.51	4.12	4.51	4.31
7%	Random	2.96	2.22	3.15	2.22	1.85	2.59	2.78	2.59	3.15	2.04
	Average	1.48	2.59	2.41	1.85	2.22	2.22	1.67	3.15	2.41	2.59
	Bandwagon	5.00	4.44	4.81	4.44	4.26	4.81	5.00	3.89	3.70	3.89
8%	Random	2.70	1.80	3.42	1.98	2.70	2.34	1.98	2.70	3.24	2.52
	Average	1.98	3.06	2.16	1.44	3.60	3.06	2.70	1.98	3.42	3.06
	Bandwagon	4.32	4.50	4.68	4.50	5.23	4.50	4.50	4.86	4.50	4.68
9%	Random	3.08	3.08	3.08	2.56	3.08	3.76	3.42	3.25	2.91	3.25
	Average	2.91	2.22	2.56	2.74	2.91	2.56	3.25	2.22	2.91	3.42
	Bandwagon	5.98	4.79	3.93	4.44	4.96	5.30	4.96	4.96	5.13	4.27
10%	Random	3.33	3.17	3.17	3.83	3.50	3.17	2.33	2.67	3.67	3.00
	Average	3.00	2.83	3.00	3.33	2.33	3.17	3.00	3.17	3.83	2.83
	Bandwagon	4.33	4.50	4.83	5.17	4.67	4.50	4.50	4.50	4.17	4.83

**Table 7** recRate of GTD model using Cosine correlation coefficient

	%	Test1	Test2	Test3	Test4	Test5	Test6	Test7	Test8	Test9	Test10
5%	Random	0.81	1.01	1.21	0.40	1.62	1.21	0.61	1.41	0.61	1.82
	Average	1.01	1.41	1.62	0.81	0.40	1.21	0.61	0.61	1.21	0.61
	Bandwagon	0.81	0.81	1.82	1.41	0.81	1.62	0.81	0.81	1.01	1.41
6%	Random	1.57	0.39	0.98	1.37	0.78	1.18	1.57	1.96	0.59	1.18
	Average	0.39	0.98	0.78	0.78	1.18	0.78	0.39	0.98	0.39	0.78
	Bandwagon	1.96	1.37	2.16	1.57	1.96	2.55	1.18	1.18	1.76	0.98
7%	Random	1.48	1.67	2.22	1.30	1.11	1.11	1.30	2.04	1.85	1.11
	Average	0.56	0.93	1.30	0.93	0.74	1.11	1.30	1.67	0.93	1.67
	Bandwagon	1.48	2.41	1.85	2.04	2.04	1.85	2.22	1.67	1.67	1.67
8%	Random	1.98	1.08	1.98	0.90	1.44	1.26	0.90	1.98	1.62	1.44
	Average	0.90	0.90	1.08	1.26	1.62	2.16	1.26	0.72	1.98	1.80
	Bandwagon	2.16	1.26	2.16	1.44	1.98	1.80	1.98	2.34	1.98	1.80
9%	Random	2.05	2.39	1.54	1.71	2.05	2.56	2.05	2.22	1.88	2.74
	Average	1.37	0.85	1.03	1.54	1.54	1.88	1.88	1.54	1.03	1.20
	Bandwagon	2.39	2.56	1.54	2.22	2.05	2.22	2.39	2.22	2.91	1.88
10%	Random	2.67	1.83	2.00	2.00	2.17	2.00	1.50	1.50	2.00	1.83
	Average	1.67	1.33	1.50	1.67	1.50	1.17	1.83	2.33	1.83	1.50
	Bandwagon	2.50	2.50	2.00	2.33	2.50	2.17	1.17	2.33	2.00	3.00

**Table 8** recRate of GTD model using Pearson correlation coefficient

	%	Test1	Test2	Test3	Test4	Test5	Test6	Test7	Test8	Test9	Test10
5%	Random	0.20	0.40	0.61	0.40	0.61	0.00	0.40	0.40	0.00	0.40
	Average	0.40	0.40	0.00	0.61	0.20	1.21	0.00	0.00	0.20	0.00
	Bandwagon	1.01	1.01	1.82	1.21	0.81	1.41	0.61	1.01	1.21	1.41
6%	Random	0.39	0.00	0.00	0.20	0.20	0.20	0.00	0.20	0.39	0.39
	Average	0.39	0.59	0.39	0.20	0.39	0.59	0.20	0.20	0.39	0.39
	Bandwagon	1.57	1.37	1.96	1.18	1.57	2.35	1.18	1.18	1.76	0.78
7%	Random	0.56	0.56	0.19	0.56	0.74	0.56	0.37	0.37	0.37	0.37
	Average	0.19	0.56	0.19	0.19	0.00	0.37	0.37	0.19	0.00	1.11
	Bandwagon	1.30	2.04	2.04	1.67	1.85	1.67	2.22	1.48	1.85	1.48
8%	Random	0.36	0.54	0.36	0.54	0.54	0.18	0.36	0.36	0.72	0.00
	Average	0.72	0.72	0.54	0.36	1.08	0.36	0.90	0.18	1.26	0.90
	Bandwagon	2.34	1.62	1.98	1.26	1.98	1.80	1.80	2.52	1.98	1.80
9%	Random	0.34	0.68	0.51	0.34	0.34	0.34	0.68	0.34	0.17	0.17
	Average	0.51	0.51	0.17	0.68	0.85	0.17	0.51	0.34	0.17	0.68
	Bandwagon	2.39	2.39	1.37	2.05	2.22	2.39	2.22	2.22	2.22	2.05
10%	Random	0.50	0.33	0.33	0.50	0.50	0.83	0.00	0.83	0.33	0.50
	Average	0.17	0.33	0.33	0.67	0.33	0.33	0.33	0.50	0.67	0.33
	Bandwagon	2.00	2.33	1.67	2.00	2.17	2.17	1.00	2.00	2.33	2.67

**Funding** Foundation of Guizhou Provincial Key Laboratory of Public Big Data No. 2017BDKFJJ015.

## Declarations

**Conflict of interest** The author declares no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will

need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Faridani V, Jalali M, Jahan MV (2017) Collaborative filtering-based recommender systems by effective trust. *Int J Data Sci Anal* 3:297–307
- Golbeck J (2009) Trust and nuanced profile similarity in online social networks. *ACM Trans Web* 3:1–33
- Ziegler C, Golbeck J (2007) Investigating interactions of trust and interest similarity. *Decis Support Syst* 43:460–475
- Hallikainen H, Laukkanen T (2018) National culture and consumer trust in e-commerce. *Int J Inf Manag* 38:97–106
- Wang Y, Qian L, Li F, Zhang L (2018) A comparative study on shilling detection methods for trustworthy recommendations. *J Syst Sci Syst Eng* 27:458–478
- Zheng XL, Chen CC, Hung JL (2015) A hybrid trust-based recommender system for online communities of practice. *IEEE Trans Learn Technol* 8:345–356
- Deng S, Huang L, Xu G (2017) On deep learning for trust-aware recommendations in social networks. *IEEE Trans Neural Netw Learn Syst* 28:1164–1177
- Borzymek P, Sydow M, Wierzbicki A (2009) Enriching trust prediction model in social network with user rating similarity. In: *International Conference on Computational Aspects of Social Networks*, Fontainebleau, France, 24–27 June, pp 40–47
- Guo G, Zhang J, Zhu F, Wang X (2017) Factored similarity models with social trust for top-n item recommendation. *Knowl Based Syst* 122:17–25
- Ma H, Lyu MR, King I (2009) Learning to recommend with trust and distrust relationships. In: *Recsys Proceedings of the Third Acm Conference on Recommender Systems*, New York, USA, October 23–25, pp 189–196
- Lee J, Lebanon G, Singer Y (2013) Local low-rank matrix approximation. In: *The 30th international conference on machine learning*, New York, USA, May, pp 741–749.
- Mackey LW, Jordan MI, Talwalkar A (2011) Divide-and-conquer matrix factorization. *Adv Neural Inf Process Syst*, pp 1134–1142
- O'Mahony M, Hurley N, Kushmerick N (2004) Collaborative recommendation: a robustness analysis. *ACM Trans Inter Tech* 4:344–377
- Mehta B, Hofmann T, Fankhauser P (2007) Lies and propaganda: detecting spam users in collaborative filtering. *International Conference on Intelligent User Interfaces*. Hawaii, USA, January 28–31, pp 14–21
- Alostad JM (2019) Improving the shilling attack detection in recommender systems using an SVM Gaussian mixture model. *J Inf Knowl Manag* 18:1–18
- Samaiya N, Raghuwanshi SK, Pateriya RK (2019) Shilling attack detection in recommender system using PCA and SVM Emerging Technologies in data mining and information security. Springer, Singapore, pp 629–637
- Lam SK, Riedl J (2004) Shilling recommender systems for fun and profit. *The 13th international conference on World Wide Web*. New York, USA, May 17–20, pp 393–402
- Ardissone L, Mauro N (2020) A compositional model of multi-faceted trust for personalized item recommendation. *Expert Syst Appl* 140:1–67
- Jaehoon L, Giseop N, Hayoung O (2018) Trustor clustering with an improved recommender system based on social relationships. *Inf Syst* 77:118–128
- Pan Y, He F, Yu H (2018) An adaptive method to learn directive trust strength for trust-aware recommender systems. In: *22nd international conference on computer supported cooperative work in design*, Nanjing, China, May 9–11, pp 10–16
- Gambetta D (2000) Trust: making and breaking cooperative relations. *Br J Sociol* 13:213–237
- McKnight DH, Chervany NL (1996) *The meanings of trust*. Springer, New York
- Lathia N, Hailes S, Capra L (2008) Trust-based collaborative filtering. In: *IFIP international conference on trust management*, Trondheim, Norway, June 18–20
- Shambour Q, Lu J (2012) A trust-semantic fusion-based recommendation approach for e-business applications. *Decis Support Syst* 54:768–780
- Guo G (2013) Integrating trust and similarity to ameliorate the data sparsity and cold start for recommender systems. In: *The 7th ACM conference on recommender systems*, HongKong, China, October 12–16, pp 451–454
- Jøsang A, Ismail R, Boyd C (2007) A survey of trust and reputation systems for online service provision. *Decis Support Syst* 43:618–644
- Peng W, Xin B (2019) SPMF: a social trust and preference segmentation-based matrix factorization recommendation algorithm. *EURASIP J Wirel Commun Netw* 1:1–12
- Cerutti F, Kaplan LM, Norman TJ (2015) Subjective logic operators in trust assessment: an empirical study. *Inf Syst Front* 17:743–762
- Bo Z, Zhang H, Li M (2017) Trust Traversal: a trust link detection scheme in social network. *Comput Netw* 120:105–125
- Papagelis M, Plexousakis D, Kutsuras T (2005) Alleviating the sparsity problem of collaborative filtering using trust inferences. *Trust management*, Paris, France, May 23–26, pp 224–239
- Sotos A, Vanhoof W, Noortgate W, Onghena P (2009) The transitivity misconception of pearson's correlation coefficient. *Stat Educ Res J* 8:33–55
- Hwang CS, Chen YP (2007) Using trust in collaborative filtering recommendation. In: *New trends in applied artificial intelligence*, Kyoto, Japan, June 26–29, pp 1052–1060
- Rahman A, Hailes S (2000) Supporting trust in virtual communities. In: *The 33rd annual Hawaii international conference on system sciences*, Hawaii, USA, January
- Hurley N, Cheng M, Zhang M (2009) Statistical attack detection. *RecSys'09*, New York, USA, October, pp 149–156
- Cosley D, Lam SK, Albert I (2003) Is seeing believing? How recommender system interfaces affect users. In: *ACM SIGCHI Int. Conf. on Human Factors in Computing Systems*, Florida, USA, April, pp 585–592
- Victor P, Cornelis C, Cock MD (2011) *Trust networks for recommender systems*. Atlantis Press, France, pp 91–107

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.