LETTER

Detection of Trust Shilling Attacks in Recommender Systems

Xian CHEN[†], **Xi DENG**[††], **Chensen HUANG**[††], ***Nonmembers***, ***and*** **Hyoseop SHIN**[†a)], ***Member***

SUMMARY Most research on detecting shilling attacks focuses on users' rating behavior but does not consider that attackers may also attack the users' trusting behavior. For example, attackers may give a low score to other users' ratings so that people would think the ratings from the users are not helpful. In this paper, we define the trust shilling attack, propose the behavior features of trust attacks, and present an effective detection method using machine learning methods. The experimental results demonstrate that, based on our proposed behavior features of trust attacks, we can detect trust shilling attacks as well as traditional shilling attacks accurately.

key words: *trust shilling attacks, recommender systems, shilling attacks detection*

1. Introduction

The recommender system has achieved great success in information filtering and has solved information overload problem efficiently. However, due to its dependence on users' historical information, some abnormal users use cheating methods to inject attacking users' information into the recommender system and manipulate these users to simulate the ratings and comments of normal users to increase or decrease the recommended frequency of the target products. This behavior is referred to as a "shilling attack." The offensive behavior interferes with the normal results of the recommender system, harms the fundamental interests of normal users, and reduces the recommendation experience of normal users and the quality of the recommendation, thereby endangering the security and robustness of the recommender system.

Scholars have conducted research and exploration in the field of shilling attacks, especially with detection. They have achieved encouraging results [1]–[5] in studies based on classification models [2]–[4], semi-supervised learning algorithms [5]–[8], unsupervised models [9], [10], and feature selection [11], [12].

In addition to giving ratings to items directly, users can give scores to or like/dislike other users' ratings and reviews. Users' scores to ratings or users' clicked like/dislike to reviews represent users' trust of ratings/reviews. Shilling attackers not only attack target items directly, such as giving direct ratings, but also attack high/low ratings. For instance, attackers may give low scores to normal users' high ratings for target items or high scores to users' low ratings and reviews. If many users give dislikes to a review of one item, it would significantly reduce the trust of the users who receive the recommendation information on this item, undermining the validity and credibility of the recommender system.

It is crucial to detect trust-based shilling attacks. We define this attack—giving low scores to normal users' high ratings and giving high scores to normal users' low ratings—as a trust shilling attack. However, few studies are related to trust-based shilling attacks. Hence, in this paper, we define trust shilling attacks, extract trusted-based features, and propose a machine learning algorithm to detect trust shilling attacks.

The rest of this paper is organized as follows. In Sect. 2, we define a trust shilling attack. In Sect. 3, we introduce our methodology, and in Sect. 4, we present the experimental results. Section 5 gives the conclusion of the paper.

2. Trust Shilling Attack

2.1 Definition of Trust Rating

Many online sites provide users not only ratings for items directly but also ratings of other users' ratings/reviews. The ratings for others' ratings/reviews reveal to what extent a user trusts the ratings/reviews of other users. Hence, we define giving ratings to others' ratings/reviews as the trust rating.

2.2 Definition of Trust Shilling Attack

We define any of the following cases to be a trust shilling attack:

- Attackers give low scores to normal users' high ratings for target items
- Attackers give high scores to normal users' low ratings for target items
- Attackers give high scores to their peers' high/low ratings for target items
- high/low ratings for competitors' items
- Attackers give low scores to normal users' high ratings for competitors' items
- Attackers give high scores to normal users' low ratings for competitors' items

Manuscript received October 14, 2021.
Manuscript revised January 10, 2022.
Manuscript publicized March 2, 2022.

[†]The authors are with Dept. of Computer Engineering, Konkuk University, Seoul, Korea.
[††]The authors are with Chongqing University of Posts and Telecommunications, China.
a) E-mail: hsshin@konkuk.ac.kr

DOI: 10.1587/transinf.2021EDL8094

- Attackers give high scores to their peers' high/low ratings for competitors' items

2.3 Behavioral Characteristics of Trust Shilling Attacks

• Attack Camouflage: For trust shilling attacks, attackers not only attack target items' rating/reviews but also give random/average scores to ratings/reviews of other non-related items (e.g., popular items) to disguise themselves. We refer to this characteristic as the *camouflage* of trust shilling attacks.
• Attack Coordination: Coordination of trust shilling attack is not just a particular attacker attacking a specific target, but a group of accomplices attacking it. Users usually give high scores to their peers' trust ratings to persuade other users to believe their trust ratings. We refer to this characteristic as the *coordination* of trust shilling attacks.
• Attack Simultaneity: If normal users give low scores to target items to which attackers want to give high scores, trust shilling attackers not only give high scores to normal users' high ratings, they also give low scores to normal users' low ratings. If normal users give high ratings to target items to which attackers want to give low scores, trust shilling attackers not only give high scores to normal users' low ratings, they also give low scores to normal users' high ratings. When attackers give both high and low scores while performing trust attacking, we refer to this characteristic as the *simultaneity* of trust shilling attacks.

3. Methodology

3.1 Extracting Trust Attack Features

• Trust similarity between trust givers and trust receivers (*TSGR*): In the trust social network, users not only give their trust to other users' ratings/reviews but also receive others' trust of their own ratings/reviews. We define *TSGR* to calculate the similarity between users' trust givers and users' trust receivers. We think normal users' *TSGR* could differ from that of trust shilling attackers. The similarity between trust givers and trust receivers of attackers may be higher than normal users because attackers may give and take trust more frequently than normal users. In Eq. (1), $TSGR_i$ is user i's trust similarity between trust givers and trust receivers. tg_i is the number of trust givers of user i, tr_i is the number of trust receivers of user i, the numerator is the overlap number of trust givers and trust receivers for user i, and the denominator is the union number of trust givers and trust receivers for user i.

$$TSGR_i = \frac{tg_i \cap tr_i}{tg_i \cup tr_i} \tag{1}$$

• Related Suspicious Frequency (*RSF*): This feature describes how many times a user gives a rating to suspicious items. In order to evaluate the RSF value for each user, we first propose how to identify suspicious items. The attacker may focus on attacking the target item for a specific period. The distribution of ratings for normal items differs from those of attacked (i.e., suspicious) items during a specific period. Thus, we calculate the similarity for each item pair using dynamic time warping [19] to determine the suspicious items. We establish a two-dimensional time-series *TS* of each item, where x is the time stamp, and y is the number of ratings received at that time, as depicted in Eq. (2). (x_n, y_c) indicates that, for time-series from x_1 to x_n, item i receives y_c times of ratings.

$$TS_i = ((x_1, y_1), (x_2, y_2) \ldots (x_n, y_c)) \tag{2}$$

Second, to calculate the similarity for each item pair (i, j), we create time-series pair (TS_i, TS_j). We regularize the path that starts at coordinates (1, 1) and ends at $(|TS_i|, |TS_j|)$, so for the path-regulated distance matrix $D(s, t)$, we produce Eq. (3).

$$D(s,t) = dist(s,t) + min\{D(s-1,t), D(s,t-1)\} \tag{3}$$

where $dist(s, t)$ represents the Euclidean distance between the s-th point in the $|TS_i|$ sequence and the t-th point in the $|TS_j|$ sequence, $D(|TS_i|, |TS_j|)$ is the final regularization path of item i and j. The smaller the value of $D(|TS_i|, |TS_j|)$, the higher the similarity between the two time-series for items i and j.

If the length of $|TS_i|$ differs from $|TS_j|$, we use k to indicate the final stretched length of the two sequences depicted as Eq. (4).

$$max(|TS_i|, |TS_j|) \leq k \ll |TS_i| + |TS_j| \tag{4}$$

As we assume that a small portion of items are suspicious and most of items are normal, items with a high average distance would be considered suspicious.
• Trust Behavior Ratio (*TBR*): normal users do trust ratings objectively. They give their trust based on their true feelings. However, trust shilling attackers may only consider attacking target items, either giving positive trust to their partners or normal users who have the same rating as them or negative trust to normal users who have different ratings/reviews. Due to the purpose and offensive nature of the trust shilling attackers, the positiveness/negativeness of attackers' trust behavior differs from that of normal users. We define Positive Trust Behavior Ratio (*PTBR*) and Negative Trust Behavior Ratio (*NTBR*) to describe this characteristic. Equation (5) illustrates how we calculate the value of *PTBR* for user u, N_u is the total count of user u giving trust to other users' ratings/reviews, and pn_u is to how many items user u gave positive trust ratings. Similarly, Eq. (6) describes how to calculate the value of *NTBR* for user u. nn_u is the total number of users u giving negative trust ratings.

$$PTBR_u = \frac{pn_u}{N_u} \tag{5}$$

$$NTBR_u = \frac{nn_u}{N_u} \tag{6}$$

3.2 Detecting Process

We verify whether our supposed characteristics can detect

trust shilling attacks by first injecting different sizes of attackers and simulating a trust shilling attack. We then extract the supposed characteristics, *TSGR*, *RSF*, and *TBR*. Finally, with these extracted features, we apply different machine learning models to detect trust shilling attacks, distinguishing trust shilling attackers and normal users.

4. Experiments

4.1 Experimental Data Set and Setting

We use the public CiaoDVD [20] data set because it has user-movie rating data and user-user trust data, which can be exposed to the trust shilling attack. The information on the data we used is presented in Table 1. The trust ratings is when, after users provide ratings to movies, other users rate those ratings.

We verify the detection effect of our schema by testing different sizes of trust shilling attackers. We separately inject 1%, 3%, and 5% of attack users and use random attacks to target items. After simulating the trust shilling attack, we use 70% data for training and 30% for testing.

Finally, we verify whether our proposed method can detect a traditional shilling attack by simulating a trust shilling attack, a traditional shilling attack, and a mixed shilling attack in our experiments.

Table 1 Information on experimental data set

Data type	Numbers
number of users	7,615
number of movies	16,121
number of ratings	72,665
number of trust ratings	1,625,480

4.2 Trust Attack Features Analysis

We evaluate the efficiency of each proposed feature by drawing normal users/attackers distributions with each feature's value, as depicted in Fig. 1. We calculate how many normal users/attacker for each value of every characteristic. The overlap means the number of normal users/attackers with same value of one characteristic. We illustrate a normal user (blue)/attacker (red) distribution with the trust-based characteristics *TSGR*, *RSF*, *PTBR* and *NTPF*. Each feature may give a clue how to recognize attackers against normal users. For example, *TSGR* can visually separate normal users and attackers by its values: attacks have a tendency of having higher values than normal users do.

In order to detect trust shilling attackers with high accuracies, we use machine learning models with these proposed features.

4.3 Experimental Results

We evaluated our detecting algorithm using precision, recall, and F1-score as metrics. Table 2 presents the results of our experiments with different learning models, attack types, and sizes. We apply 10-folds cross validation to evaluate our models, which divides our dataset into 10 folds, nine for training and one for testing. We use average F1-scores to prove our method is reliable. For trust shilling attacks, we obtained an approximately 95% ~ 98% F1-score with varying attacking sizes and by different ensemble models.

We also used our proposed trust attack features to de-

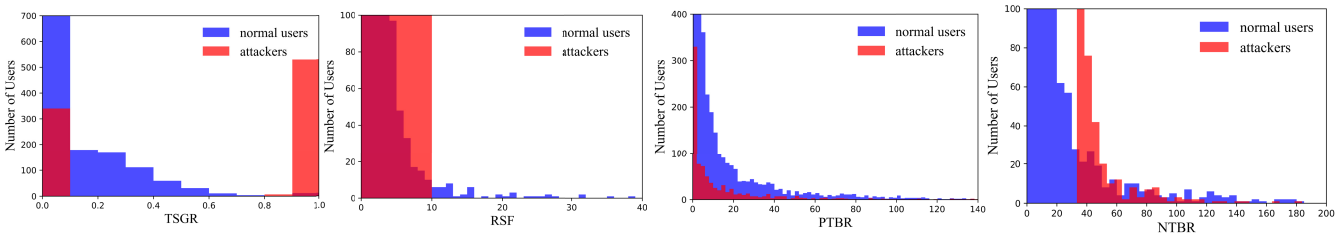

Fig. 1 Normal-attackers distributions for each feature

Table 2 F1-score of 10-folds cross-validation in detecting trust shilling attacks

Attack type/Attack Size / *Learning model*	*Trust attack*			*Shilling attack*			*Mixed attack*		
	1%	*3%*	*5%*	*1%*	*3%*	*5%*	*1%*	*3%*	*5%*
Adaboost	0.9825	0.9981	0.9780	0.9531	0.9889	0.8865	0.9888	0.9981	0.9534
GBDT	0.9855	0.9981	0.9809	0.9519	0.9869	0.8885	0.9888	0.9981	0.9524
SVM	0.9021	0.9666	0.9522	0.8568	0.9543	0.8812	0.8994	0.9651	0.9050
XGBoost	0.9856	0.9971	0.9781	0.9673	0.9879	0.8943	0.9853	0.9971	0.9440
DecisionTree	0.9771	0.9971	0.9698	0.9488	0.9802	0.8749	0.9917	0.9971	0.9387
Compared Methods	*1%*	*3%*	*5 %*	*1%*	*3%*	*5%*	*1%*	*3%*	*5%*
DSA-AURB	0.8362	0.3284	0.3412	0.7854	0.3612	0.3425	0.8655	0.3316	0.3789
FAP	0.4453	0.4289	0.4012	0.8630	0.8748	0.6968	0.4340	0.4377	0.4518
Codetector	0.8754	0.8602	0.8714	0.8174	0.8327	0.8289	0.8222	0.8128	0.6982

tect traditional shilling attacks, obtaining average F1-scores of approximately 87% ∼ 95% for the varying experimental settings.

Finally, we experimented on mixed attacks for the proposed features. The F1-score was lower than the trucst shilling attacks but higher than traditional shilling attacks. We obtained a nearly 98% F1-score of attacking sizes 1% and 3%. For the 5% size, we obtained a 94% F1-score.

In addition, we verified the performance of state-of-the art methods on conventional shilling attacks in the same experimental environments. Table 3 summarizes the result. For trust attacks, the F1-score of DSA-AURB [10] and FAP [17] were 10% ∼ 20% less than our proposed methods in average. For mixed attacks, DSA-AURB and FAP received at most 70% ∼ 80% F1-score.

5. Conclusion

In this paper, we introduced the concept of a trust shilling attack. To address this problem, we proposed several trust-attack features, including *TSGR*, *RSF*, and *TBR* which can be used as input features in arbitrary machine learning models to detect trust shilling attacks. We set up our experiment on the CiaoDVD data set. The results illustrate that our proposed method can effectively detect trust shilling attacks as well as traditional shilling attacks, while the performance of previous traditional shilling detection methods is limited.

Acknowledgments

This research was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2019R1F1A1063698) and Scientific and Technological Projects of Chongqing Education Committee (No. KJQN201900626).

References

[1] R. Burke, B. Mobasher, C. Williams, and R. Bhaumik, "Classification Features for Attack Detection in Collaborative Recommender Systems," Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.542–547, 2006.
[2] Z. Yang, Z. Cai, and X. Guan, "Estimating User Behavior toward Detecting Anomalous Ratings in Rating Systems," Knowledge-Based Systems, vol.111, pp.144–158, 2016.
[3] Z. Yang, Z. Cai, and Y. Yang, "Spotting Anomalous Ratings for Rating Systems by Analyzing Target Users and Items," Neurocomputing, vol.240, pp.25–46, 2017.
[4] F. Zhang, Z. Zhang, P. Zhang, and S. Wang, "UD-HMM: An Unsupervised Method for Shilling Attack Detection based on Hidden Markov Model and Hierarchical Clustering," Knowledge-Based Systems, vol.148, pp.146–166, 2018.
[5] C.A. Williams, B. Mobasher, R. Burke, and R. Bhaumik, "Detecting Profile Injection Attacks in Collaborative Filtering: a Classification-based Approach," Proc. Knowledge Discovery on the Web International Conference on Advances in Web Mining and Web Usage Analysis, Springer-Verlag, pp.167–186, 2006.
[6] Z. Yang, L. Xu, Z. Cai, and Z. Xu, "Re-scale Adaboost for Attack Detection in Collaborative Filtering Recommender Systems," Knowledge-Based System, vol.100, pp.74–88, 2016.
[7] W. Zhou, J. Wen, Q. Xiong, M. Gao, and J. Zeng, "SVM-TIA A Shilling Attack Detection Method based on SVM and Target Item Analysis in Recommender Systems," Neurocomputing, vol.210, pp.197–205, 2016.
[8] F. Zhang and H. Chen, "An Ensemble Method for Detecting Shilling Attacks Based on Ordered Item Sequences," Security and Communication Networks, vol.9, no.7, pp.680–696, 2016.
[9] Q. Zhou, "Supervised Approach for Detecting Average Over Popular Items Attack in Collaborative Recommender Systems," IET Information Security, vol.10, no.3, pp.134–141, 2016.
[10] H. Cai and F. Zhang, "Detecting shilling attacks in recommender systems based on analysis of user rating behavior," Knowledge-Based Systems, vol.177, pp.22–43, 2019.
[11] F. Zhang and Q. Zhou, "HHT-SVM: An Online Method for Detecting Profile Injection Attacks in Collaborative Recommender Systems," Knowledge-Based Systems, vol.65, pp.96–105, 2014.
[12] Z. Wu, J. Wu, J. Cao, and D. Tao, "HySAD: A Semi-supervised Hybrid Shilling Attack Detector for Trustworthy Product Recommendation," Proc. 18th International Conference on Knowledge Discovery and Data Mining, New York, ACM, pp.985–993, 2012.
[13] O. Riaznova, "Trust and Uncertainty: How to Communicate Successfully Book Review: Gambetta D," Journal of Economic Sociology, vol.16, no.2, pp.80–89, 2015.
[14] F. Zhang, "Average Shilling Attack against Trust-Based Recommender Systems," 2009 International Conference on Information Management, Innovation Management and Industrial Engineering, pp.588–591, 2009.
[15] W. Zhou, J. Wen, Q. Qu, J. Zeng, T. Cheng, and H. Wang, "Shilling attack detection for recommender systems based on credibility of group users and rating time series," PLoS ONE, vol.13, no.5, p.e0196533, 2018.
[16] Y. Xu and F. Zhang, "Detecting shilling attacks in social recommender systems based on time series analysis and trust features," Knowledge-Based Systems, vol.178, pp.25–47, 2019.
[17] L. Yang and X. Niu, "A genre trust model for defending shilling attacks in recommender systems," Complex Intell. Syst., 2021.
[18] Y. Zhang, Y. Tan, M. Zhang, et al., "Catch the black sheep: unified framework for shilling attack detection based on fraudulent action propagation," Proc. 24th International Joint Conference on Artificial Intelligence, pp.2408–2414, 2015.
[19] https://en.wikipedia.org/wiki/Dynamic_time_warping
[20] https://guoguibing.github.io/librec/datasets.html