# Shilling Attack Models in Recommender System

Parneet Kaur
Computer Science & Engg. Department
Thapar University
Patiala, India
parneetkaur147@gmail.com

Shivani Goel
Computer Science & Engg. Department
Thapar University
Patiala, India
shivani@thapar.edu

*Abstract*—**Recommender systems which are based on collaborative filtering are vulnerable to "shilling attacks" due to their open nature. Shillers inject a few unscrupulous "shilling profiles" into the database of ratings for altering the system's recommendation, due to which some inappropriate items are recommended by the system. In this paper, we simulated shilling attacks namely random, average, bandwagon and segment on Movie-Lens[1] dataset, which focused on a set of users having similar interests. Biased ratings of the items are also introduced in the system. The results show that although segment attack has impact on item based collaborative filtering, still it has higher robustness than user based collaborative filtering approach.**

***Keywords— Shilling attack, collaborative filtering, shilling attack models, recommender system, prediction shift***

## I. Introduction

Recommender systems (RSs) are common in e-commerce, where recommendations of items may be helpful to a customer in finding the item of his/her interest. RS predicts the ratings that would be given to an item by a user. Recent research has examined the robustness and vulnerabilities of various collaborative filtering (CF) techniques for recommendations.

Collaborative filtering recommender system (CFRS) is vulnerable to "profile injection" or "shilling attacks" [1]. Anonymous users, who cannot be easily distinguished from genuine users, insert a huge number of pseudonomous profiles into the system with the intent of manipulating its recommendation with respect to a target item.

CF recommendation approaches are grouped into two classes: user based CF and item based CF [2-5]. CFRS is vulnerable to shilling attacks due to its openness. CF algorithm based on users gathers user profiles, that represents the predilection of different individuals and provide item recommendations and predictions based on the perspective of other like-minded profiles. If the database of a system contains data which is biased, then the attackers' profile may become peer to original users and may produce result in favor of attackers. In case of item based filtering, the items similar to target item are taken into account and predict user's ratings on these similar items. From attacker's perspective, two types of efforts are there to mount an attack. First, the amount of knowledge needed for mounting an attack. In high knowledge attacks, an attacker must know the distribution of ratings in a system. A low knowledge attack is one that doesn't require details of the system. The second aspect is effort required to add the number of profiles and ratings in the system's database to make the attack effective. However, the ratings have less importance as automated software agents can be used for inserting the ratings. Sites may employ policies that limit the speed of profile multiplication. Therefore, the attack that needs large number of injected profiles in this system is less practical than that needs less number of injected profiles.

The remaining paper is arranged as follows. In section II, we explained the work done on shilling attacks till now. Section III describes the detailed description of various attack models used in our experiments, CF algorithms and metrics used for evaluating the system. Section IV introduced extensive experiments performed and their results. In section V, conclusion of our paper is given with potential direction for future work that could be done.

## II. RELATED WORK

The term "shilling" was first given by Riedl and Lam who introduced two attack models: Random Bot and Average Bot to inject attackers' profiles into the system [1]. In this, automated CF was used for generating the recommendations. They proved that item based algorithm offers advantage over user based algorithm because in item based algorithm, attacks are not much successful in altering the system's result.Amazon.com uses item-item CF approach [4]. For finding the similar customers, cluster model is used. Its main motive is to assign target user to the cluster having most similar users. Performance and scalability of cluster model is better than traditional CF algorithms. Mobasher experimentally proved that item based approach also suffers from profile injection attacks [5]. Attacks consist of attack profiles that contain biased data associated with malicious users. It has been examined that the attacks can be mounted successfully even with the little knowledge of the system [6-8]. Segment based attacks against CFRSs has been introduced which ensures that the item pushed by the attacker would be recommended to the target users [9].

Zhang [10] examined that topic level recommendation algorithm based on trust is more secure. It incorporates topic oriented trust model into CF algorithms under average attack and he concluded that this CF algorithm has more stability than standard kNN approach under average attack. Zhang [11] proposed an average hybrid and bandwagon attack model, analyzed their effectiveness against trust-based

[1] www.grouplens.org/datasets/movielens

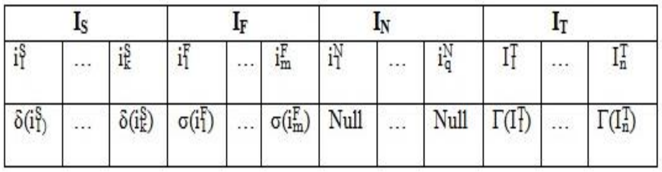
recommendation algorithm and showed that the proposed hybrid attack model has more impact on recommendations generated by system than other attack models. Donovan and Smyth introduced trust based models in CF with the aim to improve the accuracy [12]. They concluded that even trust based models are more vulnerable to shilling attack. In a later research, they modified the trust building process and solved this problem, which reduces the prediction shift by 75 percent as compared to classic CF algorithm. Calculating the similarity between the users is difficult because of sparse data in the matrix of ratings, so a trust metric is required to solve this problem. Avesani and Massa introduced a robust CF algorithm based on “web of trust” metric [13].

## III. ATTACK MODELS AND CF ALGORITHMS

### *A. Shilling Attack Models*

An attack has attack profiles which bias the recommender system’s results to their benefits, biased data and a set of target items. Shilling attacks can be categorized as nuke attack and push attack. In nuke attack, attacker gives lowest score to target items in order to demote them whereas, in push attack, attacker gives highest score to target items in order to promote them. Our main focus is on push attacks. The motive of an attacker is to construct attack profiles using attack models which have high influence and require minimum knowledge. There are four attack models that can be used: random attack, average attack, bandwagon attack, segment attack. The structure of an attacker’s profile is shown in Table I.

TABLE I. STRUCTURE OF ATTACKER’S PROFILE



| $I_S$ | | | $I_F$ | | | $I_N$ | | | $I_T$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $i_1^S$ | ... | $i_k^S$ | $i_1^F$ | ... | $i_m^F$ | $i_1^N$ | ... | $i_q^N$ | $I_1^T$ | ... | $I_n^T$ |
| $\delta(i_1^S)$ | ... | $\delta(i_k^S)$ | $\sigma(i_1^F)$ | ... | $\sigma(i_m^F)$ | Null | ... | Null | $\Gamma(I_1^T)$ | ... | $\Gamma(I_n^T)$ |

A profile for mounting an attack consists of k-dimensional ratings vector; k is the number of items in the RS. The k-dimensional vector is divided into four sets: $I_S$ , $I_F$, $I_N$, $I_T$.
The details of these four sets of items are discussed below:

- $I_S$*:* A set of randomly selected items with their ratings generated by the function $\delta(i_k^S)$.
- $I_F$*:* A set of filler items, selected randomly whose ratings are generated by the function $\sigma$ ($i_m^F$).
- $I_N$*:* A set of items those are not rated.
- $I_T$*:* A set of target items. All target items are assigned rating to maximum i.e. $\Gamma(I_n^T)=r_{max}$ (push attack) or minimum i.e. $\Gamma(I_n^T)=r_{min}$ (nuke attack).

The attack models’ features are summarized in Table II.

TABLE II. ATTACK MODELS’ FEATURES

| **Attack Models** | **$I_F$** | **$I_S$** | **$I_T$ (nuke/push)** | **$I_N$** |
|---|---|---|---|---|
| Random | overall mean | $\phi$ | $r_{min}/ r_{max}$ | $\phi$ |
| Average | item mean | $\phi$ | $r_{min}/ r_{max}$ | $\phi$ |
| Bandwagon | overall mean | $r_{min}/ r_{max}$ | $r_{min}/ r_{max}$ | $\phi$ |
| Segment | $r_{max}/r_{min}$ | $r_{min}/ r_{max}$ | $r_{min}/ r_{max}$ | $\phi$ |

#### *1) Random Attack*

It is a low knowledge attack, in which filler items ($I_F$) are selected in a random manner and rate them by using normal distribution with standard deviation and mean rating of the system. In this model, the selected item set is empty i.e. $I_S=\phi$ (null). The set of targeted items are rated with minimum or maximum depending on the type of attack i.e. nuke or push. For e.g., rating is in between 1 and 5, where 5 means liked item and 1 means disliked item, therefore, in push attack $r_{target}=5$ and in nuke attack, $r_{target}=1$. In our experiment, we have shown that this attack is not much effective in user based as well as in item based algorithm.

#### *2) Average Attack*

It is more sophisticated than other models, as described in [1]. But it is impractical to implement because it requires knowledge about the system, as it uses individual average ratings for each item instead of global mean of the system. Attackers select filler items randomly and rate the items in the database using normal distribution with mean and standard deviation of individual item. Attackers are difficult to distinguish when compared to actual users, therefore they have large impact on the system’s result. Rating pattern of targeted items is same, as in random attack. It has been shown in our experiments that, this model is not much effective. This attack model is considered as high knowledge attack as it requires the average rating of individual item. Our experiments have shown that average attack is highly effective on user based algorithm when we assigned the average ratings to a small subset of items in the database, thus reducing the knowledge requirement [5]. However, this attack is not much effective in case of item based algorithm.

#### *3) Bandwagon Attack*

In this model, attacker takes advantage of Zipf’s law distribution of popularity and generates the biased profiles that contain most popular items. Popular items are those items that are rated by lots of users. Therefore, there is a high possibility that attackers become similar to the actual users. $I_S$, a set of frequently rated items, therefore, these items together with the items in a target set, $I_T$ are assigned maximum ratings. The items in filler set are chosen in a same way as in random attack. This attack model is considered as low knowledge attack because in order to determine the popular products in any product space, knowledge required about the system is less.

#### *4) Segment Attack*

It requires less knowledge about the system. The basic concept behind this attack is to popularize the target items among a group of targeted users [9]. For e.g., an author of a romantic novel want his novel to be recommended to the readers who are the lovers of “The Notebook” (another romantic novel), not to the ones who like comics.

The fictitious user determines a set of segmented items which have high chances of being preferred by targeted users, who belong to his/her particular segment. Maximum rating is assigned to segmented items i.e. $I_S$. To maximize the attack’s impact, items in the filler set, $I_F$ are assigned ratings to the minimum, i.e. $r_{min}$=1, thus maximize the variations between the item similarities.

The attack profiles are generated using these attack models. The impact of the attack is that these attack profiles may become peer to the genuine users, manipulate the predicted ratings and their pushed item may get recommended to the users after the attack. As a result, trust on the recommender system fades away.

### *B. Recommendation Algorithms*

In our paper, we concentrated on most commonly used algorithm for predicting the ratings and recommending the items.

#### *1) User Based Collaborative Filtering*

A standard CF algorithm which finds the users who are k most similar to the targeted users and uses these users' preferences to predict the ratings is *k nearest neighbor algorithm* [2]. To find the similarity S between the users, Pearson's correlation score can be used as follows:

$$S_{a,b} = \frac{\sum_{q\in I}(r_{a,q}-\bar{r}_a)*(r_{b,q}-\bar{r}_b)}{\sqrt{\sum_{q\in I}(r_{a,q}-\bar{r}_a)^2} * \sqrt{\sum_{q\in I}^{k}(r_{b,q}-\bar{r}_b)^2}} \qquad (1)$$

where, $r_{a,q}$ and $r_{b,q}$ are the ratings given by user 'a' and its neighbor 'b' respectively, to an item q. *I* is an item set that contains all the items. The most similar users are chosen after finding the similarities between the users. Neighborhood size of 20 is taken in our experiments. The neighbors with a similarity less than or equal to zero have been rejected to prevent the negative correlations. Once the neighbors are found, rating is predicted for item q and target user 'a' by using (2):

$$P_{a,q} = \bar{r}_a + \frac{\sum_{a\in M} S_{a,b}\,(r_{b,q}-\bar{r}_b)}{\sum_{b\in M}|S_{a,b}|} \qquad (2)$$

where, $r_{b,q}$ represents users' rating who have given rating to item q, $\bar{r}_b$ denotes overall mean rating, M contains k most similar users.

#### *2) Item Based Collaborative Filtering*

This CF algorithm is based on similarities between the items [4]. kNN algorithm uses to find the k peer items. Similarities between the items are calculated using a formula, described in (1).

Once we calculate the similarities, a set of k items are selected that are most alike to the targeted item and predictions are generated using the following formula:

$$P_{a,w} = \frac{\sum_{x\in N} r_{a,x} * S_{w,x}}{\sum_{x\in N} S_{w,x}} \qquad (3)$$

where, *N* is a set of k similar items, $S_{w,x}$ represents similarity between items w and x and $r_{a,x}$ is a rating of item x predicted for user 'a'.

### *C. Evaluation Metrics*

Robustness determines how the system performs before and after the attack and how attacks affect the recommendations generated by the system. Stability measures the shift in the ratings of the pushed item before and after the attack. We measure the stability of our system via prediction shift.

*Prediction shift*

The main goal of malicious user in "push" attack is that the targeted items should have higher chances of being recommended to the target users after mounting the attack than before the attack. To compute the stability of the system, prediction shift can be used. Let *I* denotes the target items set and *U* be the set of target users. $\Delta_{a,j}$ denotes prediction shift for each pair of user and item (a,j). It can be calculated as $\Delta_{a,j}=p'_{a,j} - p_{a,j}$ , where $p'$ and *p* denotes the predicted ratings after and before the attack, respectively. The attack has been implemented successfully if the value of $\Delta_{a,j}$ is positive. For an item j over all users, average prediction shift can be evaluated as:

$$\Delta_j = \sum_{a\in U}(p'_{a,j} - p_{a,j}\,)/\,|U| \qquad (4)$$

Likewise, for all tested items, average shift in prediction can be calculated as:

$$\bar{\Delta} = \frac{\sum_{j\in I}\Delta_j}{|I|} \qquad (5)$$

Higher the value of prediction shift, higher is the chance of recommending the pushed item to the target user. But it is not true in the case, when the target item has very low scores.

## IV. EXPERIMENTAL RESULTS

### *A. Dataset Description*

MovieLens-100k dataset has been used in our experiments which contains 1,00,000 ratings rated by 943 viewers on 1682 movies. The integer value of ratings varies from 1 to 5 where 1 is assigned to most disliked items and 5 to most liked items. The dataset contains those users who rated at least 20 movies. We randomly selected 50 movies such that rating distribution for these movies is similar to the overall distribution of ratings. 63 users are taken for testing, showing the overall distribution of users as per the ratings provided. Attack is performed on each movie individually. We have used the prediction shift in order to calculate the effectiveness of these models. Usually, the values of this metric and the attack size i.e. percentage of number of biased profiles injected into the database, are plotted.

### *B. Evaluation Metric*

The attack models have been used to implement the attack and the prediction shift has been chosen to depict the robustness of our system. The value of prediction shift will indicate whether the attack is mounted successfully or not. More is the value of this measure, more is the possibility of the target item to be recommended to the users.

### C. *Experimental Setup*

In kNN algorithm, neighborhood size, k=20 has been used in our experiments. The attack models, filler size and attack size that we have used in our experiments are given below:

- *Attack models*: random, average and bandwagon
- *Filler size:* percentage of filler items, fixed at 50%.
- *Attack size:* 5%, 10%, 15%, 20%, 25%.

The details of attack profiles are described below:

- $I_T$: we randomly selected 10 unpopular items and assigned them maximum rating i.e. $r_{max}$=5.
- $I_F$: set of filler items, random ratings are assigned to items which centered around standard deviation=1.1, mean=3.6
- $I_S$: (only in bandwagon attack) selected items, first 10 most frequently rated items, were assigned to $r_{max}$, i.e. $r_{max}$=5.

Research study in [1] concluded that random and average attacks are effective against user based CF but they have less impact on item based CF. Fig. 1 and Fig. 2, show that the prediction shift of average attack is highest in almost all cases, making average model most effective and is difficult to detect. Fig. 3 shows that the average attack is strongest attack as the value of prediction shift of average attack is higher in case of item based CF than that of user based CF. Therefore, former CF algorithm has more security than later. Also, we can say that, bandwagon attack which requires less knowledge about the system, is comparable to average attack.

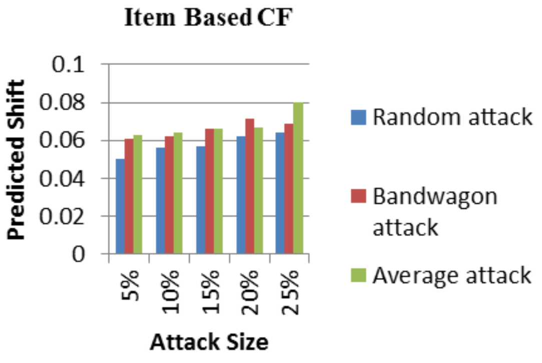


Fig.1. Attack models in item based CF.

A set of segmented users and items were identified in segment attack. Five popular horror movies[2] from the dataset as one segment were selected. Users, who gave rating greater than 3 to any three of the 5 horror movies, were chosen. Then we took the combination of those three movies that had minimum 30 users. 10 users were chosen randomly and averaged the result. Then we generated the attack profiles which were put into the system and generated the predictions.

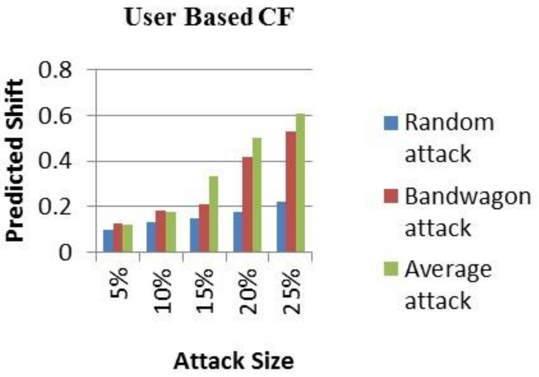


Fig.2. Attack models in user based CF.

Prediction shift of in-segment users and all users in item basis algorithm and user basis CF algorithm is shown in Fig. 4 and Fig. 5 respectively, which represents that segment attack is less effective against all users than in-segment users in both the cases. The segment attack for all users is less powerful than average attack introduced by [1] as shown in Fig. 6. Also, Fig. 7 confirms that in-segment attack in user based CF is more effective than that attack in item based CF.

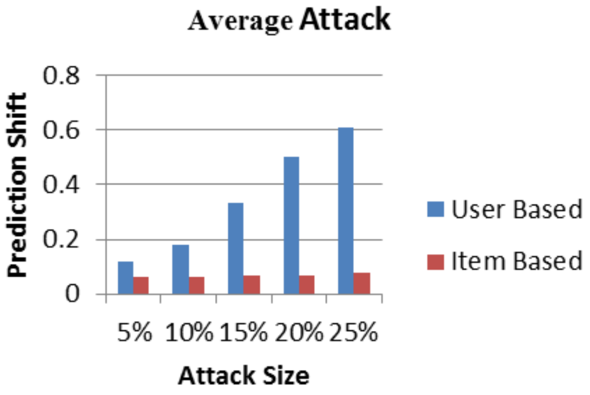


Fig. 3. Average attack in item based and user based CF.

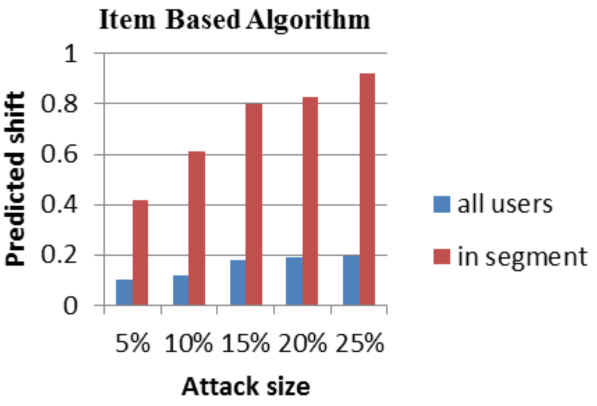


Fig. 4. Segment attack in item based CF.

[2] Horror movies are: Alien, Frighteners, Heavy Metal, The puppet Masters, From Dusk Till Dawn.

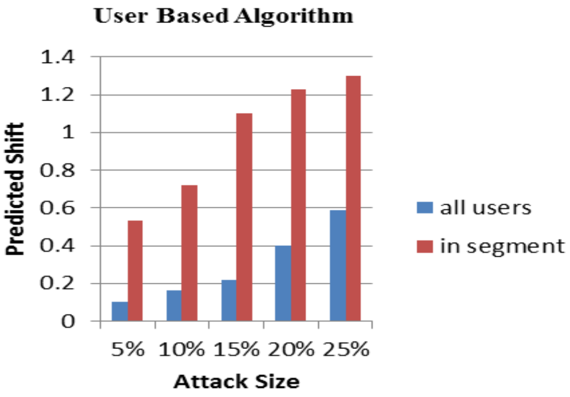


Fig. 5. Segment attack in user based algorithm.

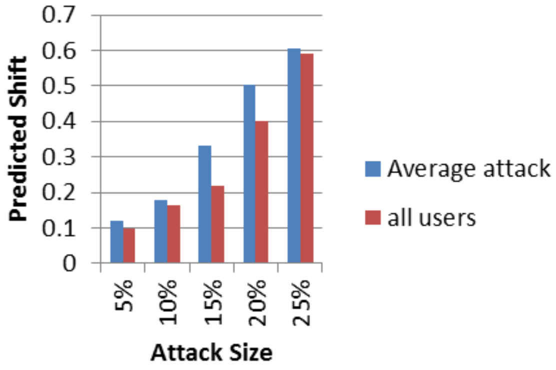


Fig. 6. Comparison in user based algorithm.

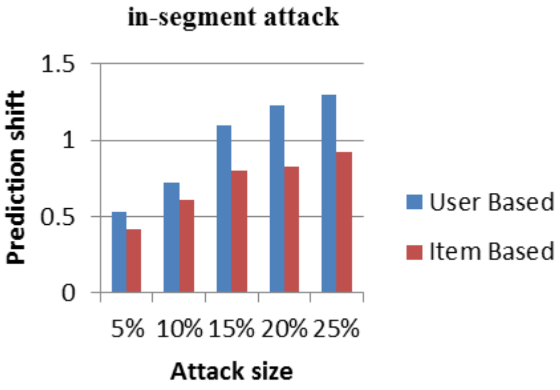


Fig. 7. Comparison of in-segment.

## V. CONCLUSION AND FUTURE SCOPE

Previous research has examined that average attack has the highest impact on the recommender system. But, it is less effective against item based algorithm and also requires more knowledge about the system. In our paper, we studied the segment attack and our results show that segment attack effects the item based algorithm to a degree that other attacks are not. But, it has more impact on user based than item based CF algorithm. Therefore, we can conclude that item based CFRSs have higher security than user based CFRSs. As a future work, hybrid models can be built to inject anonymous profiles into the system and more metrics can be considered to measure the stability.

## REFERENCES

[1] S. Lam, "Shilling Recommender Systems for Fun and Profit", *In Proceedings of the 13th international conference on World Wide Web,ACM*, pp. 393-402, 2004.

[2] J. Herlocker, J. Konstan, A. Borchers and J. Riedl, "An algorithmic framework for performing collaborative filtering", *In Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, 1999.

[3] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Item-Based Collaborative Filtering Recommendation", *ACM, Hong Kong*, pp. 285-295, 2001.

[4] G. Linden, B. Smith and J. York, "Amazon.com recommendations: item-to-item collaborative filtering", *IEEE Internet Computing*, vol. 7, no. 1, pp. 76-80, 2003.

[5] B. Mobasher, R. Burke, R. Bhaumik and C. Williams, "Effective Attack Models for Shilling Item-Based Collaborative Filtering Systems", *In Proc. of the 2005 WebKDD Workshop, Chicago, Illinois*, 2005.

[6] M. Mahony, N. Hurley and C. Silvestre, "Recommender Systems: Attack Types and Strategies", *American Association for Artificial Intelligence*, pp. 334-339, 2005.

[7] B. Mobasher, R. Burke, R. Bhaumik and J. Sandvig, "Attacks and Remedies in Collaborative Recommendation", *IEEE Intell. Syst.*, vol. 22, no. 3, pp. 56-63, 2007.

[8] M. O'Mahony, N. Hurley, N. Kushmerick and G. Silvestre, "Collaborative recommendation: A robustness analysis", *ACM Transactions on Internet Technology*, pp. 344-377, 2004.

[9] R. Burke, B. Mobasher, R. Bhaumik and C. Williams, "Segment-based injection attacks against collaborative filtering recommender systems", *In Data Mining, Fifth IEEE International Conference*, 2005.

[10] 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.

[11] F. Zhang, "Analysis of Bandwagon and Average Hybrid Attack Model against Trust-based Recommender Systems", *Fifth International Conference on management of e-Commerce and e-Government*, pp. 269-273, 2011.

[12] J. O'Donovan and B. Smyth, "Trust in Recommender Systems", *IUI, Association for Computing Machinery, New York, NY, USA*, 2005.

[13] P. Massa and P. Avesani, "Trust-aware recommender systems", *in Proceedings of the 1st ACM Conference on Recommender Systems (RecSys '07)*, pp. 17-24, 2007.