Clustering Approach to detect Profile Injection Attacks in Recommender System

Ashish Kumar
CSED, Thapar University,
Patiala, India

Deepak Garg
CSED, Thapar University,
Patiala, India

Prashant Singh Rana
CSED, Thapar University,
Patiala, India

ABSTRACT
Recommender systems apply techniques of knowledge discovery for specific problem to make personalized recommendation of the products or services to the users. The huge growth in the information and the count of visitors to the web sites especially on e-commerce in last few years creates some challenges for recommender systems. E-commerce recommender systems are vulnerable to the profile injection attacks, involving insertion of fake profiles into the system to influence the recommendations made to the users. Prior work has shown that performance of system can be affected by even small number of biased profiles. In this paper, we show that unsupervised clustering approach can be used effectively for the detection of profile injection attacks in recommender system. Here we give a comparative study of four clustering algorithms and measure their performance.

Keywords
Recommender system; collaborative filtering; attack detection; bias profile injection; performance measure; unsupervised approach.

1. INTRODUCTION
Recommender system predicts the preference that a user would give to an item. Many websites uses these recommender systems that provides its users a list of items or web pages that are likely to interest them. Recommender systems have been developed to a variety of applications like movies, music, books, news, research articles or other products.

Content based and collaborative filtering based are two widely used types of recommender systems. Content based filtering approach uses the properties of an item to recommend additional items with similar properties [1]. Collaborative filtering works by building a database of preferences for various items in the system by users [2]. It works on the basic principle that two users who had similar tastes in the past will also have similar taste in future also providing that tastes do not changes vary rapidly. Among these two approaches of recommendation, for e-commerce recommender systems, collaborative filtering approach is most common. A popular well known example of recommender system is Amazon.com [4] and it uses item to item collaborative filtering based recommender system.

As collaborative filtering based system is open to the users input, so they have high chances of attacks often termed as "shilling" attacks [3, 5, 6] or "profile injection" attacks. The main aim of the attacker is to interact with the RS to build a large number of fake user profiles in the system with the target of affecting the system output i.e. either push (promote) or nuke (demote) a particular item [7]. Previous work done on has shown that if recommender systems of e-commerce are not protected from these attacks, there is a very high risk that the trust of customers in the site and predictions can be affected by the attacker.

2. RELATED WORK
The term shilling was used first time in [6]. For the detection of profile injection attacks lots of researches has been carried out [8, 10]. Attack detection algorithms have been categorized in supervised, unsupervised and semi-supervised.

Supervised technique has been used in [9] for the detection of shilling attacks. [11] Uses three classification algorithms SVM, C4.5 and kNN to improve the robustness of the system. Labeled dataset is required for supervised algorithms to improve the accuracy. More effort is required for these techniques because first training data is prepared and then algorithm is evaluated on test data.

The third category, semi-supervised uses both labeled as well as unlabeled dataset. For hybrid attack detection Wu et al. [17] proposed a system HySAD. It is semi supervised learning system that uses both labeled and unlabeled user profiles for multi class modeling. Hurley et al. [16] uses Neyman-Pearson theory to develop both supervised and unsupervised detectors.

3. BACKGROUND
User-user based collaborative filtering also called kNN collaborative filtering. It was first of the automated collaborative method. User-user collaborative filtering use a straightforward approach based on the concept of collaborative filtering. Many user based systems like GroupLens [20], Ringo [18] and BellCore video [19] evaluate the interest for an item i by user u, using the ratings by other users called neighbors that have similar rating pattern or similar interests. To identify "Usenet" articles that are likely to be interesting to a specific user GroupLens [21] used this approach. Users are required to provide ratings and system combines these ratings with the ratings provided by other users to generate personalized results and users need not know the opinion of the other users. To make prediction for a user first its similarity is calculated with other user, Pearson correlation is widely used for this purpose then based on this similarity, and prediction is made for that user.

$$S_{u,v} = \frac{\sum_{i \in I}(r_{u,i} - \overline{r_u})(r_{v,i} - \overline{r_v})}{\sqrt{\sum_{i \in I}(r_{u,i} - \overline{r_u})^2}\sqrt{\sum_{i \in I}(r_{v,i} - \overline{r_v})^2}} \quad (1)$$

Where $S_{u,v}$ is the similarity between user u and v, $u, v \in U$. $i \in I$ is the subset of items rated by both users u and v. $r_{u,i}$ Denotes the rating given by the user u to item i and $\overline{r_u}$ is the average of all ratings given by user u [22]. Both GroupLens and BellCore used Pearson correlation to compute the similarity in their projects [19, 21]. By selecting the items with maximum ratings, recommendations are generated.

$$P_{u,i} = \frac{\sum_{n \in Neighbors}(r_{n,i} - \overline{r_n})S_{u,n}}{\sum_{n \in Neighbors}|S_{u,n}|} + \overline{r_n} \quad (2)$$

4. PROFILE INJECTION ATTACKS

In this section, we present two main aspects of profile injection attacks that must be analyzed: attack models and attack dimensions.

A. Attack models

A profile injection attack in an RS have a set of profiles injected in the system by the attacker. Each profile having four set of items: a single target item i_t, a set of selected items I_S based on the properties of the attack, a set of randomly chosen filler items I_F and $I_\emptyset$ a set of items that are unrated. A general structure of these profiles is shown in Fig. 1.

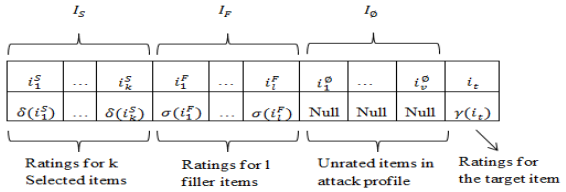

Fig 1. General structure of a profile in a profile injection attack.

Lam and Riedl [2004] [6] introduced two basic types of attack models i.e. random and average attack models. For push attack highest rating is given to the i_t (target item) i.e. $r_{target} = r_{max}$ and for nuke attack least rating is given to i_t (target item) i.e. $r_{target} = r_{min}$. In the random attack average rating of system is assigned to I_F and highest rating is given to the target item and I_S is kept empty. It is very easy to implement but it has limited effectiveness. The average attack is very similar to it except mean rating of individual item is assigned to the items belonging to I_F. Average attack may be impractical to mount because it requires knowledge of ratings in the system and random attack is less effective than average attack [24]. Bandwagon attack [25] is same as random attack model but it needs some more knowledge to find the popular items in the systems. The set I_S contains these popular items. Maximum rating is assigned to the items belonging to this set along with target item. In reverse bandwagon attack minimum rating is assigned to set I_S along with target item and average rating of system is assigned to I_F.

B. Attack dimensions

An attack can be categorized based on the size of the attack, the intent of the attack and knowledge needed by the attacker to inject the attack. From the attackers' perspective, best attack is one that requires least effort and yield maximum impact. From the perspective of detection we are interested to know how these factors are combined to form the dimension of attack. Primary dimensions of the attacks are:

i. **Knowledge required:** it depends on the attack model; some attacks require more system knowledge than others. Higher knowledge requirement makes difficult to implement the attack.
ii. **Attack intent:** it describes the intention of the attacker, whether the attacker wants to push (promote) or nuke (demote) an attack.
iii. **Profile size:** number of ratings given by a specific attack profile is called profile size. Although cost of assigning ratings is lower as compared to creating a new profile. But a large profile size is a good indicator of attack profile because real user rarely gives ratings to the large number of items.
iv. **Attack size:** it tells about the number of attack profiles inserted by the attacker in the system. To increase the cost of profile insertion registration or capcha is used, which require human intervention.

5. PROFILE INJECTION ATTACKS

Detection attributes are divided into two types: generic and model specific attributes. Generic attributes relies on the overall signature of attack profile that makes it different from genuine profile. Model specific attribute are used to detect attack profile based on the specific characteristics of known attack. Generic attributes are less effective as compared to the model specific attributes.

A. Generic attributes for detection

i. **Rating deviation from mean agreement (RDMA):** t_i is the number of ratings given for an item by all the users, N_u be the number of item rated by user u [23].

$$RDMA_u = \frac{\sum_{i=0}^{N_u} \frac{|r_{ui} - \bar{r}_i|}{t_i}}{N_u} \quad (3)$$

ii. **Weighted deviation from mean agreement (WDMA):**

$$WDMA_u = \frac{\sum_{i=0}^{N_u} \frac{|r_{ui} - \bar{r}_i|}{|t_i|^2}}{N_u} \quad (4)$$

iii. **Degree similarity with top neighbors:**

$$DegSim_u = \frac{\sum_{v=1}^{k} sim_{uv}}{k} \quad (5)$$

iv. **Length Variance:** average length of profile denoted by $\bar{l}$, l_u is the length of profile u.

$$LengthVar_u = \frac{|l_u - \bar{l}|}{\sum_{k \in U} (l_k - \bar{l})^2} \quad (6)$$

B. Model specific attributes

i. **Mean variance:** for average attack this metric is used. $|P_u|$ Is the number of ratings in profile P_u . And $\bar{r}_i$ is mean rating of item i across all users.

$$MeanVar_{(p_t,u)} = \frac{\sum_{i \in (P_u - P_t)} (r_{i,u} - \bar{r}_i)^2}{|P_u|} \quad (7)$$

ii. **Filler mean target difference (FMTD)** used for bandwagon attack. Set of all items in P_u that are assigned highest rating in user u's profile denoted by $P_{u,T}$ and all other items in P_u become the set $P_{u,F}$.

$$FMTD_u = \left(\frac{\sum_{i \in P_{u,T}} r_{u,i}}{|P_{u,T}|}\right) - \left(\frac{\sum_{k \in P_{u,F}} r_{u,k}}{|P_{u,F}|}\right) \quad (8)$$

6. EXPERIMENTAL RESULTS

In our experiment we used the movie lens 100K dataset. This dataset contains 943 users, 1682 movies and 100,000 ratings. All the ratings are between 1 to 5 and an integer value where 1 is the minimum and 5 is the maximum rating. Ratings to at least 20 movies are given by each user. For each profile injection attack, we keep tracking filler size, attack size and compare the accuracy i.e. fake or genuine user, of different clustering approaches.

In this paper, we compare the accuracy of EM (expectation maximization), Farthest First, Hierarchical Clusterer and Simple K Mean. To test the robustness of the best predictive model we perform k fold cross validation. For our experiment, half of the attacks describe in Fig. 2, 4, 6 and 8, are inserted at fixed 10% attack size and varying filler size of 10%, 20%, 30%, 40% and 50%. For other half attacks are describe in Fig. 3, 5, 7 and 9, we insert attacks at varying size 1%, 3%, 6%, 9%, 12% and 15% and we keep filler size of 5%.

In Fig. 2, 4 and 8, Hierarchical Clusterer gives same accuracy in every case and it is independent of the filler size. In Fig. 6, accuracy of Farthest First and Hierarchical Clusterer is almost same and constant in every case. In Hierarchical Clusterer and Simple K Mean, to measure distance between two individual we use Euclidean Distance. In our experiment, we found that Hierarchical Clusterer and Farthest First are two best performing clustering approach where EM and Simple K Mean are two least performing approach.

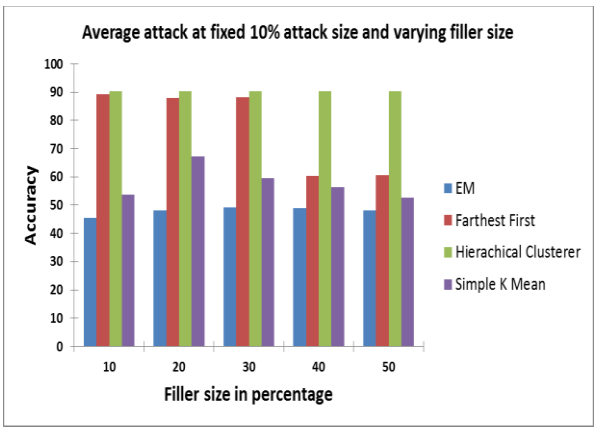

Fig 2: Performance analysis of clustering algorithms in average attack at fixed 10% attack size and varying filler size.

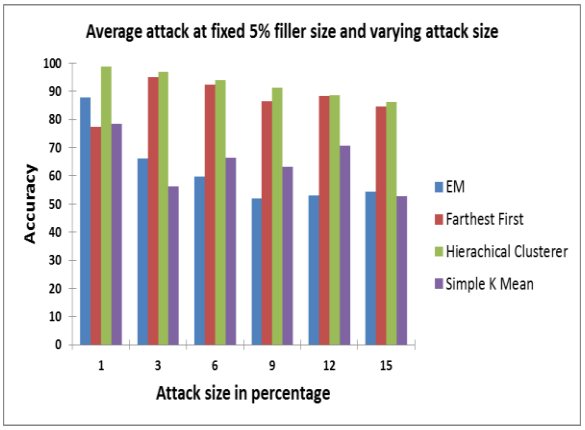

Fig 3: Performance analysis of clustering algorithms in average attack at fixed 5% filler size and varying attack size.

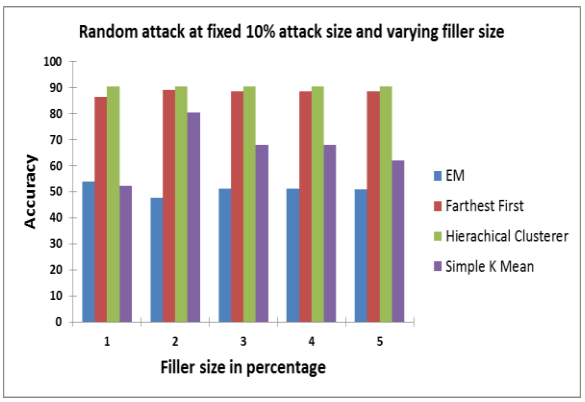

Fig 4: Performance analysis of clustering algorithms in random attack at fixed 10% attack size and varying filler size.

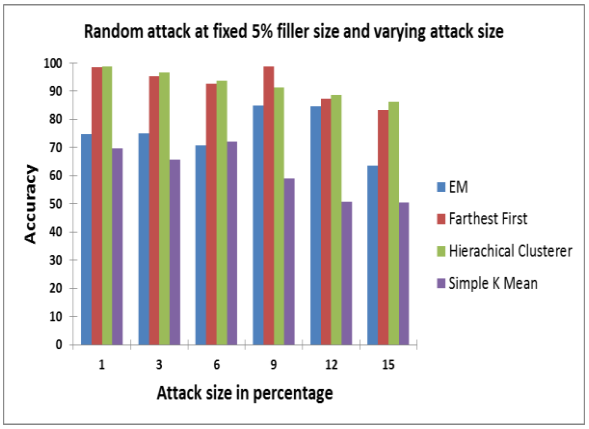

Fig 5: Performance analysis of clustering algorithms in random attack at fixed 5% filler size and varying attack size.

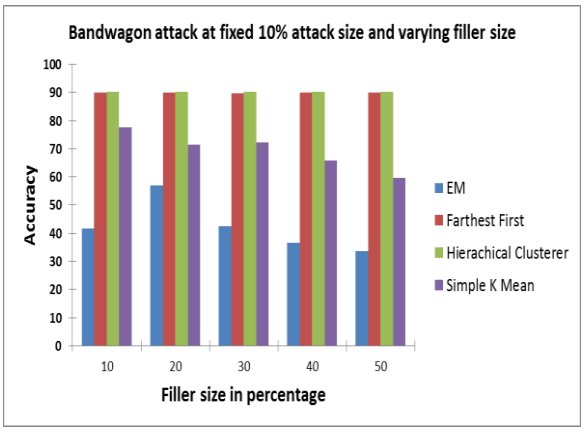

Fig 6: Performance analysis of clustering algorithms in bandwagon attack at fixed 10% attack size and varying filler size.

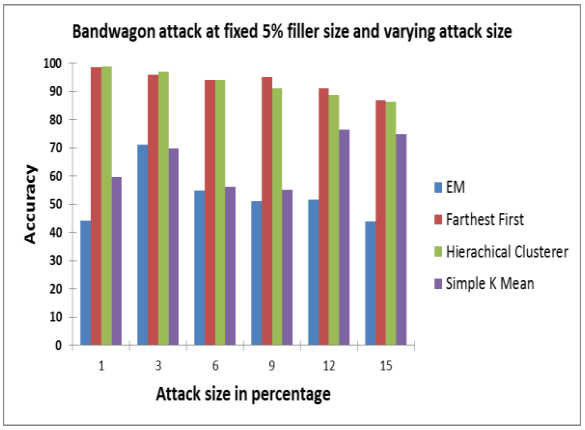

Fig 7: Performance analysis of clustering algorithms in bandwagon attack at fixed 5% filler size and varying attack size.

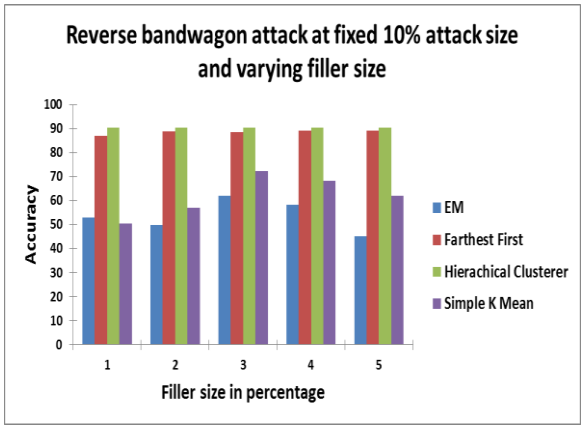

Fig 8: Performance analysis of clustering algorithms in reverse bandwagon attack at fixed 10% attack size and varying filler size.

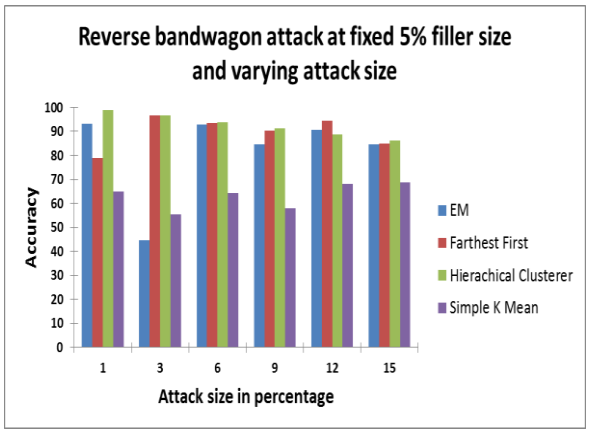

Fig 9: Performance analysis of clustering algorithms in reverse bandwagon attack at fixed 5% filler size and varying attack size.

K-fold cross validation is used to evaluate the accuracy of our predictive models. We randomly partition our sample into 10 subsamples. Out of these 10 subsamples, one sample is used for testing and remaining 9 subsamples are used for training. This process is repeated 10 times with each of the 10 samples used exactly once for validation. After this 10 results from the folds are averaged for single estimation value. The main advantage of this process is that all the values are used for both validation and training.

7. CONCLUSION AND FUTURE WORK

In this work, we examine four machine learning clustering approaches and measure their performance for the detection of attack profiles in recommender system. Based on their performance we find out that Hierarchical Clustering approach is best performing approach with accuracy above 80% in almost every case. EM clustering approach is least performing approach in most of the case on our dataset. All the modules are evaluated on RDMA, WDMA, degree similarity, length variance and model specific attributes. We perform 10-fold cross validation to calculate the robustness of all the four clustering approaches. It is expected that optimization of model parameters may leads to better results. This approach can be used in other areas also like spam filtering, intrusion detections etc.

8. REFERENCES

[1] Davoodi, Fatemeh Ghiyafeh, and Omid Fatemi. "Tag based recommender system for social bookmarking sites." In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, pp. 934-940. IEEE Computer Society, 2012.

[2] Bobadilla, Jesús, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. "Recommender systems survey." *Knowledge-Based Systems* 46 (2013): 109-132.

[3] Burke, Robin, Bamshad Mobasher, Roman Zabicki, and Runa Bhaumik. "Identifying attack models for secure recommendation." In *Beyond Personalization: A Workshop on the Next Generation of Recommender Systems*. 2005.

[4] Linden, Greg, Brent Smith, and Jeremy York. "Amazon.com recommendations: Item-to-item collaborative filtering." *Internet Computing, IEEE* 7, no. 1 (2003): 76-80.

[5] O'Mahony, Michael, Neil Hurley, Nicholas Kushmerick, and Guénolé Silvestre. "Collaborative recommendation: A robustness analysis." *ACM Transactions on Internet Technology (TOIT)* 4, no. 4 (2004): 344-377.

[6] Lam, Shyong K., and John Riedl. "Shilling recommender systems for fun and profit." In *Proceedings of the 13th international conference on World Wide Web*, pp. 393-402. ACM, 2004.

[7] Burke, Robin, Bamshad Mobasher, Runa Bhaumik, and Chad Williams. "Segment-based injection attacks against collaborative filtering recommender systems." In *Data Mining, Fifth IEEE International Conference on*, pp. 4-pp. IEEE, 2005.

[8] Herlocker, Jonathan L., Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. "Evaluating collaborative filtering recommender systems." *ACM Transactions on Information Systems (TOIS)* 22, no. 1 (2004): 5-53.

[9] Burke, Robin, Bamshad Mobasher, Chad Williams, and Runa Bhaumik. "Classification features for attack detection in collaborative recommender systems." In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 542-547. ACM, 2006.

[10] O'Mahony, Michael P., Neil J. Hurley, and Guénolé Silvestre. "Detecting noise in recommender system databases." In Proceedings of the 11th international conference on Intelligent user interfaces, pp. 109-115. ACM, 2006.

[11] Williams, Chad A., Bamshad Mobasher, and Robin Burke. "Defending recommender systems: detection of profile injection attacks." Service Oriented Computing and Applications 1, no. 3 (2007): 157-170.

[12] Fu, Lin, Dion Hoe-Lian Goh, Schubert Shou-Boon Foo, and Jin-Cheon Na. Collaborative querying through a hybrid query clustering approach. Springer Berlin Heidelberg, 2003.

[13] Lee, C-H., Y-H. Kim, and P-K. Rhee. "Web personalization expert with combining collaborative filtering and association rule mining technique." Expert Systems with Applications 21, no. 3 (2001): 131-137.

[14] Zhang, Sheng, Yi Ouyang, James Ford, and Fillia Makedon. "Analysis of a low-dimensional linear model under recommendation attacks." In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 517-524. ACM, 2006.

[15] Mehta, Bhaskar, and Wolfgang Nejdl. "Unsupervised strategies for shilling detection and robust collaborative filtering." User Modeling and User-Adapted Interaction 19, no. 1-2 (2009): 65-97.

[16] Hurley, Neil, Zunping Cheng, and Mi Zhang. "Statistical attack detection." In Proceedings of the third ACM conference on Recommender systems, pp. 149-156. ACM, 2009.

[17] Fu, Lin, Dion Hoe-Lian Goh, Schubert Shou-Boon Foo, and Jin-Cheon Na. Collaborative querying through a hybrid query clustering approach. Springer Berlin Heidelberg, 2003.

[18] Shardanand, Upendra, and Pattie Maes. "Social information filtering: algorithms for automating "word of mouth"." In Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 210-217. ACM Press/Addison-Wesley Publishing Co., 1995.

[19] Hill, Will, Larry Stead, Mark Rosenstein, and George Furnas. "Recommending and evaluating choices in a virtual community of use." In Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 194-201. ACM Press/Addison-Wesley Publishing Co., 1995.

[20] Konstan, Joseph A., Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. "GroupLens: applying collaborative filtering to Usenet news." Communications of the ACM 40, no. 3 (1997): 77-87.

[21] Resnick, Paul, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. "GroupLens: an open architecture for collaborative filtering of netnews." In Proceedings of the 1994 ACM conference on Computer supported cooperative work, pp. 175-186. ACM, 1994.

[22] Su, Xiaoyuan, and Taghi M. Khoshgoftaar. "A survey of collaborative filtering techniques." Advances in artificial intelligence 2009 (2009): 4.

[23] Chirita, Paul-Alexandru, Wolfgang Nejdl, and Cristian Zamfir. "Preventing shilling attacks in online.

[24] Kumar, Ashish, Deepak Garg, and Prashant Singh Rana. "Ensemble approach to detect profile injection attack in recommender system." Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on. IEEE, 2015.

[25] Noh, Giseop, Young-myoung Kang, Hayoung Oh, and Chong-kwon Kim. "Robust Sybil attack defense with information level in online Recommender Systems." *Expert Systems with Applications* 41, no. 4 (2014): 1781-1791.