Published Online 02-March-2021

IMPACT ANALYSIS OF PROFILE INJECTION ATTACKS IN RECOMMENDER SYSTEM

Ashish Kumar[1], Yudhvir Singh[2], Vikas Siwach[3], Harkesh Sehrawat[4*]
*1,2,3,4*UIET, Maharshi Dayanand University, Rohtak, India*
Email: sehrawat_harkesh@yahoo.com

Abstract: Recommender systems are the backbone of all the prediction-based service platforms e.g. Facebook, Amazon, LinkedIn etc. Even companies now a days are using the recommender systems to show users personalized ads. These service providers capture the right audience for their services/ products and hence, improve overall sales. Social networking platforms are using recommender systems for connecting people of similar interests which is almost impossible without recommender systems. Collaborative filtering-based recommender system is most widely used recommender system. It is used in this research to predict the rating for a specific movie. Accuracy of the prediction define the performance of the overall system. The quality of predictions is degraded by the attackers by injection of fake profiles. In this paper, the various types of profile injection attacks are explained and the attack scenario gets extended to measure the performance of these attacks. Empirical results on the real world publicly available data set shows that these attacks are highly vulnerable. The impact of these attacks in several conditions has been measured and it is tried to find the scenarios where these attacks are more powerful.

Keyword: Recommender system, collaborative filtering, profile injection attacks, prediction shift.

Introduction

Now a day's, recommender systems (RSs) are widely used in almost all online personalized service platforms. Internet is full of information and content. Finding the right thing is very challenging. For example, a user wants to buy a jean on amazon. Before purchasing the item, first he will go through the product description, read the comments/reviews by the other users. However, buying the product like this is very challenging and difficult. Here comes RS in the picture. RS is just like an online salesman. It filters out the suitable item for the user from the huge database based upon the interest of the user. It takes care of the several parameters like ratings, choices, popularity, availability etc. Based on all these parameters it filtered out the most relevant items for the user. For example, amazon uses the RS to recommend new items it may be clothing, books movies etc., Facebook uses it to shows the advertisement, friend suggestions, news feeds etc., LinkedIn uses it to recommend the jobs, connections etc. Similarly, there are hundreds/thousands of other platforms also which uses RS to provide the personalized services to their customers/users. In the simplest form, personalized recommended items are shown in the form of ranked list of items and this ranking is done by the RSs based upon the user's preferences and constraints. User's preference is collected from his past interaction with the system or filters that he applied while interaction with the web portal of the service provider [1].

Collaborative filtering (CF) based RS are most widely used. Which is based on the concept that if two user had similar tastes in the past then chances are that they may have similar tastes in the future also [2, 20]. CF based RS are of two types. In first one, correlation is calculated between users which is known as user-user based CF, in second one, correlation is calculated between the items which is known as item-item based CF. RS used by amazon.com is based on the item-item based CF [3]. RSs plays vital role in several online portals like Facebook, Amazon, YouTube, Netflix, Yahoo, IMDb etc. Netflix an online movie rental service provider offers million dollars to the team who improved the performance of their RS [4]. Content based filtering approach is another popular type of RS. It uses the properties of the item to categorize the items. For example, e-newspapers uses this type of RS [8]. There are several other types of recommender system also like knowledge based, hybrid RSs etc. But in this paper, focus is on the CF based RSs.

As the performance of CF based RSs is highly dependent on other users. So, these RSs are highly vulnerable to the attacker [5, 6]. These attacks are known as "Shilling Attacks" or "Profile Injection Attacks". Main idea behind these attacks is that attacker create fake user profiles and inject in the system with an aim to promote (push) or demote

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

(nuke) a specific or group of specific items [7]. In this paper, various types of the attacks are and the capability of the attack profiles to measure their performance in different scenarios is extended. The performance of the attack is measured in term of the prediction shift.

Literature Survey

Collaborative filtering-based recommender systems are widely used by the online platforms. Tapestry, a mail filtering system for the intranet at Xerox Palo Alto Research Center was the one of the earliest CF based RS [9]. In CF, to predict the rating that a user can give to unseen item is calculated by two step process. First it needs to calculate the similarity between the users. This can be done using the either Pearson correlation or cosine similarity. In this paper, the Pearson correlation is used. It can be calculated by the equation 1.

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

Where $S_{a,b}$ is the similarity between user a and b; $a, b \in U$. U is the set of all the users. $i \in I$ where i is the set of items rated by both the users a and b and I is the set of all the items. $r_{a,i}$ denotes the rating given by user a to item i. $\bar{r}_a$ denotes the average of all the ratings given by user a [10]. Similarity between two users is a value between -1 and 1. Where 1 is the maximum similarity and -1 is the minimum similarity. BellCore and GroupLens used the Pearson correlation in their project [11, 12]. Once the similarity between users is calculated, then based upon their similarity prediction is made by using the equation 2 [13].

$$P_{a,i} = \frac{\sum_{n \in N}(r_{n,i}-\bar{r}_n)(S_{a,n})}{\sum_{n \in N}|S_{a,n}|} + \bar{r}_n \quad (2)$$

Where $P_{a,i}$ is the prediction for the user a of the item i. N is the set of all the users who has maximum similarity with user a.

a. Attack Profile Structure

RSs generates the recommendation based upon the interaction of the user with the system. Attackers take the advantage of this loophole by injecting bogus/fake users in the system. These fake users behave in such a manner that they become difficult to identify [14]. Idea behind these fake users is to promote (push) or demote (nuke) some particular item/items [15]. Based upon the rating pattern, set of items is divided into four categories as shown in fig.1.

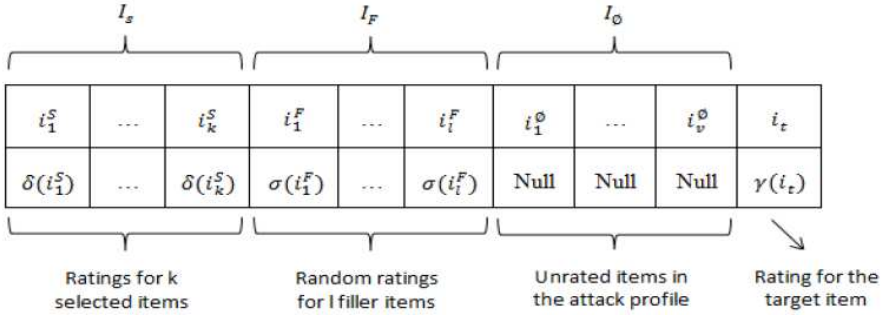


Fig. 1: Attack profile structure.

Where i_t is the target item/items that the attacker wants to promote or demote in case of push attack maximum possible rating is given to i_t and minimum rating is given in case of nuke attack. $I_s$is the set of selected items that is chosen based upon the their association with the target item. I_F is the set of filler items that is chosen randomly and rating to these items are given based upon the type of attack. Remaining set of items $I_\emptyset$ is unrated [16, 24].

b. Profile Injection Attacks

Lam and Riedl introduces average and random attack [17]. In random attack, average rating of the system is given to the set of filler items. In average attack, average rating of the item is given the items belonging to the set of filler items. Maximum rating is given to the target item/items and set of selected items is not given any rating in both average and random attack [18]. Bandwagon attack needs more knowledge because it selects most popular items in the set of selected items and rest is similar to random attack. Bandwagon attack also used to demote specific item/items whereby it selects least popular attacks in the set of selected items and it give least possible rating to the set of target items. Love and hate attack give minimum rating to the set of target items and gives maximum possible rating to the set of the filler items [19]. Segment attack require high knowledge about the database. Where set of selected items is chosen in such a way that these items lie in the same category of target item and maximum possible rating is given to the items of this set. It gives minimum possible rating to the set of filler items [21, 23]. Same attack can be used for both pushing and nuking a specific item by changing the rating pattern. In this research, the segment attack is not implemented.

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

Table 1. Features of profile injection attacks.

Attack Name	Intention	I_s	I_F	$I_\emptyset$	I_t
Average Attack	Push	Null	Average rating of the item belonging to I_F	Null	r_{max}
Random Attack	Push	Null	Average rating of all the items in the system	Null	r_{max}
Bandwagon Attack	Push	r_{max} is given to all the items belonging to I_s	Average rating of all the items in the system	Null	r_{max}
Segment Attack	Push	r_{max} is given to all the items belonging to I_s	r_{min}	Null	r_{max}
Reverse Bandwagon	Nuke	r_{min} is given to all the items belonging to I_s	Average rating of all the items in the system	Null	r_{min}
Love-Hate Attack	Nuke	Null	r_{max} is given to all the items belonging to I_F	Null	r_{min}

Before injecting the attacks, attackers take care of several issues such as. For example, knowledge required to mount the attack, intention behind the attack weather attacker wants to promote or demote the specific item, size of the attack profile which is determined by the numbers of rating given by the fake user and size of the attack which is determined by the number of attack profiles injecting in the system by the attacker etc. [8].

Experimental Evaluation

In order to measure the impact of attacks in the recommendations generated by the systems, several experiments on the real-world dataset are conducted.

a. Dataset

Dataset released by MovieLens in 2018 is used [22]. A detailed description of dataset is shown in the table 2. This dataset is having 100836 ratings of 9724 movies with averaging each movie is having 10 ratings. Data is having 610 users each has given average 165 ratings.

Table 2. Description of dataset.

Attribute	Value
Number of ratings	100836
Number of movies	9724
Least movie ID	1
Maximum movie ID	193609
Number of users	610
Minimum user ID	1
Maximum user ID	610
Average number of ratings by each user	165.305
Average number of ratings of each movie	10.369
Minimum possible rating	0.5
Maximum possible rating	5.0
Average rating of the movies	3.502
Median of the ratings	3.5

b. Experimental Methodology and Results

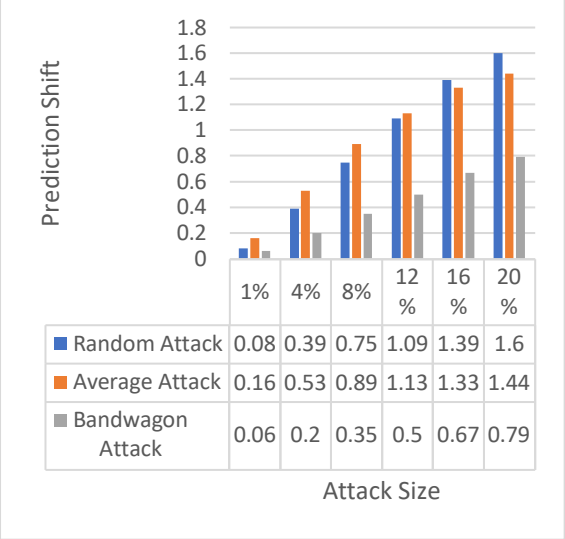
In this experiment, the impact of attacks in eight different scenarios is measured. Table 3 describes all the combinations in which this experiment is conducted.

Table 3: Profile injection attack scenario.

Attack Type	Fixed Attribute	Variable Attribute
Push Attack	Filler Size (2%) and Target Size 1	Attack Size (1%, 4%, 8%, 12%, 16%, 20%)
	Filler Size (2%) and Target Size 10	Attack Size (1%, 4%, 8%, 12%, 16%, 20%)
	Attack Size (10%) and Target Size 1	Filler Size (1%, 1.5%, 2%, 2.5%, 3%)
	Attack Size (10%) and Target Size 10	Filler Size (1%, 1.5%, 2%, 2.5%, 3%)
Nuke Attack	Filler Size (2%) and Target Size 1	Attack Size (1%, 4%, 8%, 12%, 16%, 20%)
	Filler Size (2%) and Target Size 10	Attack Size (1%, 4%, 8%, 12%, 16%, 20%)
	Attack Size (10%) and Target Size 1	Filler Size (1%, 1.5%, 2%, 2.5%, 3%)
	Attack Size (10%) and Target Size 10	Filler Size (1%, 1.5%, 2%, 2.5%, 3%)

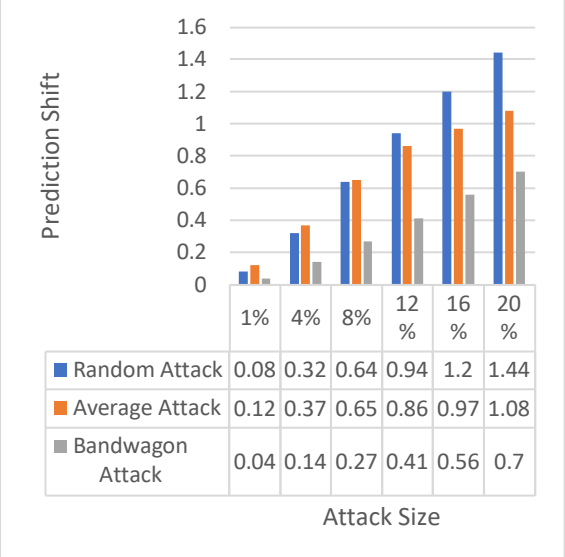
ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

For each attack type, two attributes are fixed and the third attribute is varied. The filler size is selected in such a way that it oscillates around the average rating per user in the system. As average number of rating by each user is 165. The attack profiles have given ratings to the filler movies between 97 and 291. Reason behind this is to make the attack profile difficult to identify. For push attach, target movie is chosen randomly from the pool of movies which has average rating between 2.4 and 2.5 so that significant prediction shift can be measured. Attack profiles varies from 1% to 20% as described in the table 3. For nuke attack, target movie from a pool of movies is selected which has average ratings between 4.3 and 4.5.



	1%	4%	8%	12 %	16 %	20 %
Random Attack	0.08	0.39	0.75	1.09	1.39	1.6
Average Attack	0.16	0.53	0.89	1.13	1.33	1.44
Bandwagon Attack	0.06	0.2	0.35	0.5	0.67	0.79

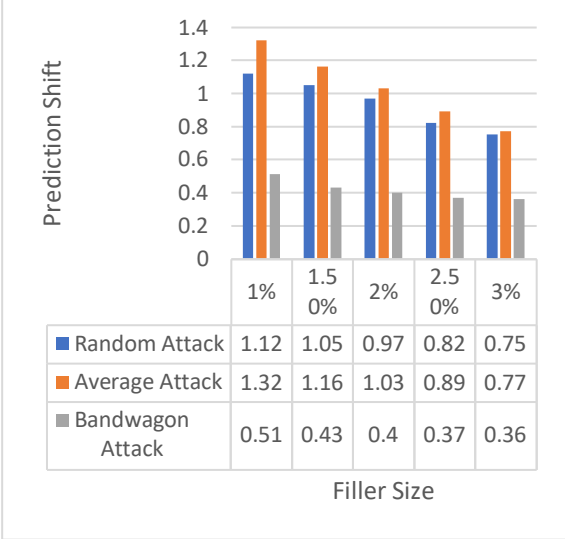
Fig. 2: Prediction shift versus attack size when filler size is fixed at 2% and target movies size is 1 (Push Attack).



	1%	4%	8%	12 %	16 %	20 %
Random Attack	0.08	0.32	0.64	0.94	1.2	1.44
Average Attack	0.12	0.37	0.65	0.86	0.97	1.08
Bandwagon Attack	0.04	0.14	0.27	0.41	0.56	0.7

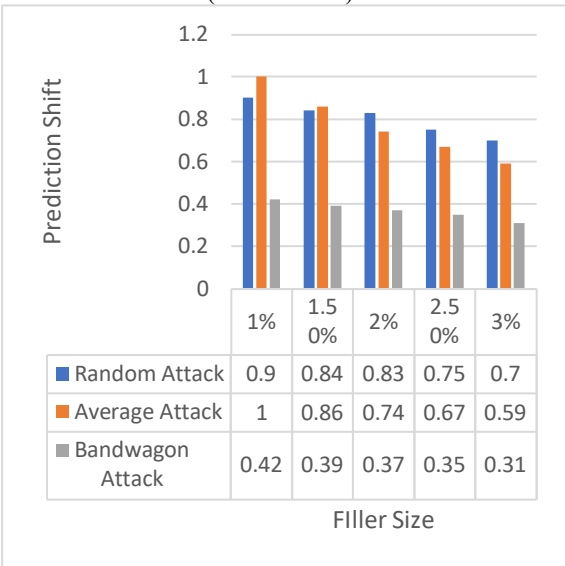
Fig. 3: Prediction shift versus attack size when filler size is fixed and target movies size is 10 (Push Attack).

Fig. 2 and Fig. 3 describes the performance of push attacks. In fig. 2, the filler size is fixed to 2% and target movie to 10 whereas attack size varies. It is found that up to the time attack size of 12%, average attack behaved more powerful as compare to other two attack but as soon the attack size is increased, random attack starts performing better than average attack where bandwagon attack performs very poorly. In Fig. 3, the target movie size is fixed to 10 and rest is same. In this case, it is found that as soon as attack size crosses 8%, random attack start performing well as compared to other attacks but before this, threshold average attack was performing better. Bandwagon attack perform poorly in this case also.



	1%	1.5 0%	2%	2.5 0%	3%
Random Attack	1.12	1.05	0.97	0.82	0.75
Average Attack	1.32	1.16	1.03	0.89	0.77
Bandwagon Attack	0.51	0.43	0.4	0.37	0.36

Fig. 4: Prediction shift versus filler size where attack size is fixed 10% and target movie size is 1 (Push Attack).



	1%	1.5 0%	2%	2.5 0%	3%
Random Attack	0.9	0.84	0.83	0.75	0.7
Average Attack	1	0.86	0.74	0.67	0.59
Bandwagon Attack	0.42	0.39	0.37	0.35	0.31

Fig. 5: Prediction shift versus filler size where attack size is fixed 10% and target movie size is 10 (Push Attack).

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

In fig. 4 and fig. 5, performance of three push attacks i.e., random, average and bandwagon are described. Here attack size is fixed at 10% and filler size is varying between 1% to 3%. In fig. 4, Target movies size is 1 and average attack perform better as compared to other attacks. Although gap in the prediction shift of average and random attack is large when filler size is less but as the filler size increases, random attack covers the gap and start performing equals to the average attack. Prediction shift does not change much as the filler size increase in case of bandwagon attack. In fig. 5, Target movies size is 10 and average attack perform better as compared to the random attack when filler size is less but as the filler size increases gradually, random attack starts performing better than average attack where bandwagon attack remains within same range only. Although in both the cases in fig. 4 and fig. 5, prediction shift of all three attacks decreases as the filler size increases.

In fig. 6 and fig. 7, prediction shift of nuke attacks i.e., love-hate and reverse bandwagon attacks are described. Here attack size is varying between 1% and 20% while filler size is kept fixed at 2%. In fig. 6, target movie size is kept 1. Love-hate attack is performing much better than reverse bandwagon attack. As the attack size increases, performance of both the attacks increases but love-hate attack keeps performing better than reverse bandwagon attack. In fig. 7, target movies size is 10 and the performance of love-hate attack is better here also as compared to reverse bandwagon attack. With the increase in attack size, performance of both the attacks is increasing but as compared to previous case in fig. 6, performance is far lower where reason might be averaging of the prediction shift.

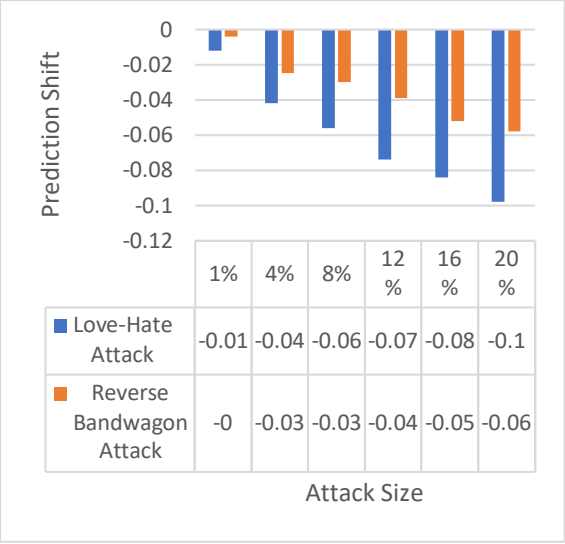


Fig. 6: Prediction shift versus attack size where filler size is fixed 2% and target movie size is 1 (Nuke Attack).

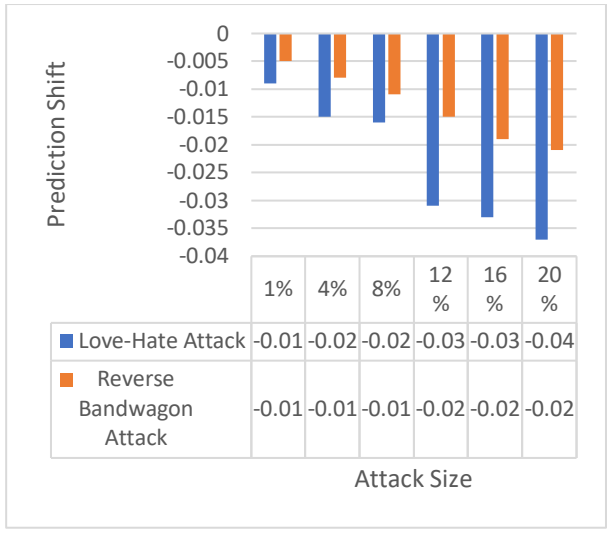


Fig. 7: Prediction shift versus attack size where filler size is fixed 2% and target movie size is 10 (Nuke Attack).

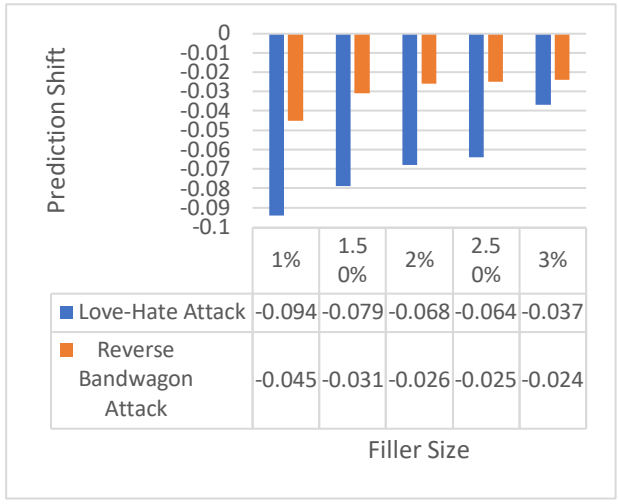


Fig. 8: Prediction shift versus filler size where attack size is fixed 10% and target movie size is 1 (Nuke Attack).

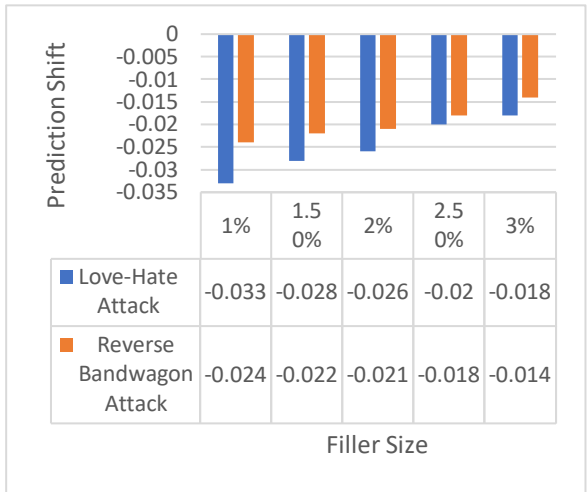


Fig. 9: Prediction shift versus filler size where attack size is fixed 10% and target movie size is 10 (Nuke Attack).

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

Fig. 8 and fig. 9, describes the performance of the nuke attacks i.e., love-hate and reverse bandwagon attacks. Here attack size is kept fixed at 10% while filler size varies between 1% and 3%. In fig. 8, target movies size is 1. Love-hate attack perform much better than reverse bandwagon attack. Initially, when filler size is less prediction shift on love-hate attack is significantly high but it started falling as the filler size increases and it falls sharply after filler size become greater than 2.5%. In fig. 9, target movies size is 10. Here also, performance of the love-hate attack is better than reverse bandwagon attack. But overall performance of both the attacks is much lower than previous case in fig. 8. Possible reason for this downtrend is averaging of prediction shift as the number of target movies increase. Similar pattern is seen in previous cases also in fig. 5 and fig. 7.

Conclusion and Future Scope

In this paper, it is derived that performance of average attack is better than other push attacks and in case of nuke attacks, performance of love-hate attack is better. Another important point observed is that as the attack size is increased, performance of all the attacks is increased. But when the filler size is increased, performance is decreasing. Another point observed is that when the target movies are more than 1, performance of all the attack whether it is push attack or nuke attack is decreasing.

This research can be carried forward to the next level by finding more types of attacks. Multiple attacks can be combined and then the performance of the attacks can be compared. Research can be done in discovering the techniques to filtered out the fake profiles. For this purpose, research can be done in finding the most accurate attributes of the attacks. So that accuracy of finding the attacks can be improved. This research can be very helpful for the ecommerce websites or social media platforms. Similar type of research can be made in the financial sector also to detect the fraud transactions.

References

1. Si, Mingdan, and Qingshan Li. "Shilling attacks against collaborative recommender systems: a review." Artificial Intelligence Review 53.1 (2020): 291-319.
2. Bobadilla, Jesús, et al. "Recommender systems survey." *Knowledge-based systems* 46 (2013): 109-132.
3. Linden, Greg, Brent Smith, and Jeremy York. "Amazon. com recommendations: Item-to-item collaborative filtering." *IEEE Internet computing* 7.1 (2003): 76-80.
4. Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." *Computer* 42.8 (2009): 30-37.
5. O'Mahony, Michael, et al. "Collaborative recommendation: A robustness analysis." *ACM Transactions on Internet Technology (TOIT)* 4.4 (2004): 344-377.
6. Lam, Shyong K., and John Riedl. "Shilling recommender systems for fun and profit." *Proceedings of the 13th international conference on World Wide Web*. 2004.
7. Burke, Robin, et al. "Segment-based injection attacks against collaborative filtering recommender systems." Fifth IEEE International Conference on Data Mining (ICDM'05). IEEE, 2005.
8. Kumar, Ashish, Deepak Garg, and Prashant Singh Rana. "Ensemble approach to detect profile injection attack in recommender system." *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2015.
9. Goldberg, David, et al. "Using collaborative filtering to weave an information tapestry." *Communications of the ACM* 35.12 (1992): 61-70.
10. Su, Xiaoyuan, and Taghi M. Khoshgoftaar. "A survey of collaborative filtering techniques." *Advances in artificial intelligence* 2009 (2009).
11. Resnick, Paul, et al. "GroupLens: an open architecture for collaborative filtering of netnews." *Proceedings of the 1994 ACM conference on Computer supported cooperative work*. 1994.
12. Hill, Will, et al. "Recommending and evaluating choices in a virtual community of use." *Proceedings of the SIGCHI conference on Human factors in computing systems*. 1995.
13. Kumar, Ashish, Deepak Garg, and P. Singh. "Clustering approach to detect profile injection attacks in recommender system." *International Journal of Computer Applications* 166.6 (2017): 7-11.
14. Burke, Robin, et al. "Identifying attack models for secure recommendation." *Beyond Personalization* 2005 (2005).
15. O'Mahony, Michael, et al. "Collaborative recommendation: A robustness analysis." *ACM Transactions on Internet Technology (TOIT)* 4.4 (2004): 344-377.
16. Turk, Ahmet Murat, and Alper Bilge. "Robustness analysis of multi-criteria collaborative filtering algorithms against shilling attacks." *Expert Systems with Applications* 115 (2019): 386-402.
17. Lam, Shyong K., and John Riedl. "Shilling recommender systems for fun and profit." *Proceedings of the 13th international conference on World Wide Web*. 2004.
18. Gunes, Ihsan, et al. "Shilling attacks against recommender systems: a comprehensive survey." *Artificial Intelligence Review* 42.4 (2014): 767-799.

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

19. Si, Mingdan, and Qingshan Li. "Shilling attacks against collaborative recommender systems: a review." *Artificial Intelligence Review* 53.1 (2020): 291-319.
20. Burke, Robin, Michael P. O'Mahony, and Neil J. Hurley. "Robust collaborative recommendation." *Recommender systems handbook*. Springer, Boston, MA, 2015. 961-995.
21. Mobasher, Bamshad, et al. "Attacks and remedies in collaborative recommendation." *IEEE Intelligent Systems* 22.3 (2007): 56-63.
22. MovieLens homepage: https://grouplens.org/datasets/movielens/, last accessed: 05/09/2020.
23. Si, Mingdan, and Qingshan Li. "Shilling attacks against collaborative recommender systems: a review." Artificial Intelligence Review 53.1 (2020): 291-319.
24. Verma, Anjani Kumar, and Veer Sain Dixit. "A Comparative Evaluation of Profile Injection Attacks." Advances in Data and Information Sciences. Springer, Singapore, 2019. 43-52.

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731