IJCSI International Journal of Computer Science Issues, Volume 14, Issue 4, July 2017
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org https://doi.org/10.20943/01201704.4450

A Comprehensive Study of Shilling Attacks in Recommender Systems

Tulika Kumari[1], Dr. Punam Bedi[2]

[1]Department of Computer Science, Assistant Professor, Shaheed Rajguru College of Applied Sciences for Women, University of Delhi, Delhi, 110096, India

[2]Department of Computer Science, Professor,University of Delhi, Delhi, 110007, India

Abstract

With the abundance of data available, it becomes difficult to distinguish useful information from massive amount of information available. Recommender systems serves the purpose of filtering information to provide relevant information to users that best acknowledge their needs. In order to generate efficient recommendations to its target users, a recommender system may use user data such as user identity, demographic profile, purchase history, rating history, browsing behavior etc. This may raise security and privacy concerns for a user. The goal of this paper is to address various security and privacy issues in a recommender system. In this paper, we also discuss some of the evaluation metrics for various attack models.
Keywords: *Privacy, security, recommender system, shilling attacks.*

1. Introduction

Recommender systems are used to aid users find relevant items. Recommender systems are a type of information filtering system that attempts to predict the preferences of a user. A variety of techniques has been proposed for generating recommendations, including content-based, collaborative, demographic, utility-based, knowledge based filtering.

In order to work accurately, recommender system often collects user's personal information. According to Ackerman, Cranor and Reagle [1] there are three main types of customers:

-Privacy fundamentalists : Users who are against any use of their personal information.

-Pragmatic majority: The pragmatics are users who are also concerned about data usage but less so than the fundamentalists. Their concerns are often reduced by the presence of some privacy protection measures.

-The **marginally concerned** users provide personal information to websites easily.

According to Shyong et al [2], there can be three types of violation of user trust:

Exposure. User trust is violated if a recommender system gives undesired access to personal information of a user.

There has been many cases of online privacy breaches in the recent times. A popular social media site, Facebook has been afflicted by some serious privacy concerns over the years. In October 2010, Facebook admitted that some of its apps shared personal data of users with advertisers. In March 2011, California-based insurer HealthNet announced a privacy breach for nearly 2 million of its customers, exposing their names, addresses, Social Security numbers, health and financial data.

Bias. This type of violation occurs when user recommendations are manipulated to alter the items that are recommended.

Biasness can affect the recommender system input which in turn may reduce the ability of a recommender system to generate accurate recommendations. Biases can distort or manipulate user preferences therefore lead to suboptimal product choices. Bias manipulates or distorts user preferences by either making an item more or less visible to the user i.e. by pushing or nuking the visibility of an item. This often reduces user trust in the recommender system and thus harm the system. Therefore, getting rid of biasness in a recommender system is an important research question.

Sabotage. User trust may be violated if a recommender system reduces its accuracy intentionally. One of the most common and oldest sabotage technique is denial of service attack. Spam link building to your website, duplication of your websites' content etc may reduce the accuracy of your recommender system.

A recommender system can gain user confidence by recommending some items that user already likes or knows. Even though, it does not add any value to the recommender system , but helps in building users' trust for other unknown items recommended to him. Giving an outline of the process of recommendation generation to user also helps in making the system more credible. Transparency in the recommender system helps in enhancing users' trust in the recommender system. Another obvious method for user trust evaluation is to ask the user to give their feedback.

In this paper we will discuss biasness in detail. This paper is designed, as follows : In section 2 we discuss shilling attacks in collaborative filtering based recommender system. Section 3 focuses on various shilling attack detection strategies and evaluation metrics are covered in section 4. Section 5 focuses on hit ratio and prediction shift of average and random attacks. Further we conclude our paper in section 6.

2. Shilling Attacks

Shilling attacks are one of the most discussed attack methods these days in which an attacker tries to generate biased recommendations for an item. In shilling attack, attacker tries to manipulate system recommendations for a particular item by submitting misrepresented opinions to the system [3].

2.1. Characterizing Attacks.

Collaborative filtering based recommender system mainly generates recommendations on the basis of user profiles i.e., it predicts interests of a user with the help of an idea that users who liked similar items in the past are likely to again agree in the future. In collaborative filtering recommender systems, an attacker can weaken the recommender system either by making a particular item look like a good recommendation for a particular user (when it is actually not a good recommendation) or by preventing an item from being recommended to a user (when it is actually a good choice for him). So, two main attack strategies widely used in recommender systems are product push and product nuke attacks. The aim of product push attack is to increase the prediction value of items being targeted by the system and product nuke attack demotes the predictions for target items. To implement these attack strategies, an attacker creates a large number of fake profiles designed to distort the system predictions. These fake user profiles are also referred to as attack profiles and insertion of attack profiles in a recommender system is referred to as **profile injection attack**.

Lam et al.[3] classified profile injection attacks on the basis of amount of knowledge required to mount the attack, intent of the attack, cost of the attack etc.

2.1.1 Required knowledge

An attack is classified into two types on the basis of amount of knowledge required to mount that attack [4].

High- knowledge attack: It requires an attacker to have complete knowledge of the procedure with which ratings are distributed in the recommender systems database.

Low- knowledge attack: This type of attack does not require detailed knowledge of rating distribution. It requires system independent knowledge that can easily be obtained by public information sources.

2.1.2 Attack intent

Another dimension used for categorizing profile injection attack is the intent of attacker. Depending upon the intent of attacker, researchers have categorized attacks into two types : "push" and "nuke" attacks.

Push attack: If an attacker inserts fake user profiles in the system with an aim to promote a product, it is known as push attack.

Nuke attack: If an attacker inserts fake user profiles in the system with an aim to demote a product, it is known as nuke attack.

2.1.4 Cost

A shill attacker on the basis of cost/benefit of an attack may determine whether it is economically good for him to execute the attack or not.

Factors that contribute to the cost of an attack are:

- size of attack: the number of new users and ratings.
- level of difficulty encountered while interacting with a recommender system. For example, an attack on a recommender

system that uses CAPTCHA for security reasons is comparatively costlier than the attack on recommender systems that do not use it.

- cost/benefit of an attack also depend upon the amount of information /knowledge such as algorithms, users, items, ratings etc required by an attacker to mount that attack.
- any other resource required to mount the attack.

2.2 Shilling attack strategies Shilling attack types are the attacks that involve injection of user profiles in the system. A general form of genuine user profile is shown in table 1.

Table 1. Genuine user profile

$Item_1$	$Item_2$	....	$Item_{m-1}$	$Item_m$
R_1	R_2		-	-

rated items unrated items

A general form of attacker's profile is shown in table 2.

Table 2. Fake user profile/ attacker's profile

selected items	filler items	unrated items	rated items

Is_1	Is_2	I_{f1}	I_{f2}	..	I_{m-1}	I_m
$\delta(Is_1)$	$\delta(Is_2)$	$\sigma(I_{f1})$	$\sigma(I_{f2})$	..	Null	¥(I_m)

ratings of selected items	ratings of filler items	unrated item	rating of target item

2.2.1 Random attack In this type of attack, attacker assigns random ratings to filler items and a pre specified rating is assigned to the target item. To implement push attack, attacker assigns maximum rating to the target item and likewise minimum rating is assigned to the target item in nuke attack strategy. This is a low knowledge attack as the amount of knowledge required to mount this attack is minimal.

2.2.2 Average attack In average attack, each filler item is assigned a rating that is mean rating for that item, across the users in the database who have rated it.

2.2.3 Bandwagon Attack In this type of attack, attacker tries to associate target item with some frequently rated items. To implement bandwagon attack, attacker usually gives random rating to a subset of items (like in random attack) and maximum rating to very popular items which in turn increases the similarity of attack profiles with other users.

2.2.4 Segment attack Segment attack was introduced by Mobasher et al.[5]. In this attack model, attacker pushes an item to its target user group that is expected to be most preferable for that group. For example, the author of a romantic novel might want to get the book recommended to the readers who liked romantic novels in the past.

2.2.5 Nuke Attack Models

Though random and average attack models can be used to demote or nuke an item by associating minimum rating instead of maximum rating with the target item. However results suggest that the attack models which are effective for pushing an item are not necessarily as effective for nuke attacks. Thus some attack models have been designed by the researchers especially for demoting an item.

2.2.5.1 Love/Hate attack

In this type of attack model, attacker gives minimum rating value to the target items and the filler items are assigned maximum rating value.

2.2.5.2 Reverse Bandwagon Attack

The reverse bandwagon attack is a variation of the bandwagon attack, in which the selected items are those that tend to be rated poorly by many users. These items are assigned low ratings together with the target item. Thus the target item is associated with widely disliked items, increasing the probability that the system will generate low predicted ratings for that item.

3. Shilling attack detection strategies

In order to mount an attack, attacker tries to gain the knowledge of ratings in the recommender system. Lam and Riedl 2004[2], Chirita et al. 2005[5] along with other researchers believe that it is impossible for an attacker to have the complete knowledge of user ratings in a recommender system. That is why fake user profiles exhibit some features that are different from that of genuine users.

3.1 Generic attributes

For the detection of profiles, Chirita et al. [6] proposed some generic attributes. Some of these attributes are as follows:

1) Rating Deviation from Mean Agreement (RDMA) : Chirita et al.[5] proposed that a profile's average deviation per item, weighted by the number of ratings for that item can be used for the detection of attacker's profile from other users. Following equation (2.3.1) can be used for this purpose:

$$\text{RDMA}_U = \frac{\sum_{i=0}^{n_u} \left| \frac{r_{u,i} - \overline{r_i}}{l_i} \right|}{n_u} \quad (2.3.1)$$

where n_u is the number of items user u rated, $r_{u,i}$ is the rating given by user u to item i, l_i is the number of ratings provided for item i by all users, and r_i is the average of these ratings.

2) Weighted deviation from mean agreement (WDMA): This method can be used to identify the user profiles that assign a high weight to rating deviations for sparse items. It can be calculated by using Equation 2.3.2.

$$\text{WDMA}_U = \frac{\sum_{i=0}^{n_u} \left| \frac{r_{u,i} - \overline{r_i}}{l_i^2} \right|}{n_u} \quad (2.3.2)$$

3) Degree of Similarity with Top Neighbors (DegSim): Chirita et al. 2005 [6] proposed that the average similarity of a profile's top nearest neighbors can be used to identify attack profiles as they are believed to have high similarity with their top 25 nearest neighbors than authenticated users [5,6]. It can be calculated by Equation 2.3.3.

$$\text{DegSim}_u = \sum_{v=1}^{k} \text{sim}_{u,v} \; / \; k \quad (2.3.3)$$

Similarity between user u and v is represented by $\text{sim}_{u,v}$.

4) Length Variance (LengthVar): According to some researchers, in a system having very large database, genuine users are unlikely to have large number of items rated/viewed by them as they have to enter the information manually. On the contrary, an attack profile may contain large profile as they often use some tool for profile injection. Mobasher et al proposed that it is possible to identify attack profiles using the variance in the length of a given profile from the average length in the database [4]. It can be calculated by using Equation 2.3.4.

$$\text{LengthVariance}_u = \frac{|l_u - \bar{l}|}{\sum_{k \in U} (l_k - \bar{l})^2} \quad (2.3.4)$$

$\bar{l}$ is the average length of profiles in the system and l_u is the length of user u in the system.

3.2 Model-specific attributes

According to Burke et al. [4]; Mobasher et al. [5] generic attributes are not very good choice for distinguishing the attack profiles from the authentic profiles. They proposed that a particular attack model has its unique signature that can be used for profile detection i.e., attacks can be characterized based on the characteristics of their partitions (target items), selected items , filler items. Some of the model-specific attributes are :

1) **Mean Variance (MeanVar)**
If the set $P_{u,T}$ contains the items in the profile that are suspected to be targets and P_u is the profile of user u. MeanVar for P_t in the profile P_u where P_t is from the set of items $P_{u,T}$ in P_u that are given the rating r_t (the maximum rating for push attack detection or the minimum rating for nuke attack detection).
It is used for the detection of average attacks [4]. It can be calculated by using Equation 2.3.5.

$$\text{MeanVar}(Pu, Pt) = \frac{\sum_{i \in (Pu - Pt)} (r_{i,u} - \overline{r})^2}{|P_u|} \quad (2.3.5)$$

2) Filler Mean Target Difference (FMTD)
This attribute is used for bandwagon, reverse bandwagon attack and segment attacks [4]. In this model, $P_{u,T}$ is set to all items in P_u that are given maximum rating for push attack detection and minimum for nuke attack detection in the profile of user 'u'. It can be calculated by using Equation 2.3.6.

$$\text{FMTD}_u = \left| \left(\frac{\sum_{i \in P_{u,T}} r_{u,i}}{|P_{u,T}|} \right) - \left(\frac{\sum_{k \in P_{u,F}} r_{u,k}}{|P_{u,F}|} \right) \right| \quad (2.3.6)$$

4. Evaluation Metrics

There are various metrics for evaluating effectiveness of an attack. We used two such metrics, prediction shift and hit ratio as described below.

4.1 Prediction Shift

Prediction shift evaluates efficaciousness of attack by computing difference in the predicted ratings for the attacked item before and after the attack.

For each user-tem pair (u,i) the prediction shift is measured as $PredShift_{u,i} = p'_{u,i} - p_{u,i}$

where $p'_{u,i}$ represents predicted rating of user-item pair(u,i) after the attack and $p_{u,i}$ represents predicted rating of user-item pair(u,i) before the attack.

Likewise, average prediction shift for an item 'i' over all users can be computed as

$$PredShift_i = \sum_{u \in U} \frac{PredShift(u,i)}{|U|} \quad (4.1.1)$$

4.2 Hit Ratio

A user is generally interested in the top *n* items recommended to him in a recommender system. A remarkable prediction shift does not guarantee presence of pushed item in recommendation list of target user. Thus, hit ratio is another extensively used evaluation metric for assessing the impact of attacks.

Let R_u be the recommendation list for target user *u*. For each pushed item *i* , the recommendation hit for user *u* on item *i*, denoted by

$$H_{u,i} = \begin{cases} 1 & ; \text{if } i \in R_u \\ 0 & ; \text{otherwise} \end{cases}$$

Hit ratio for an item *i* is defined as the ratio of number of hits across all users to total number of users.

$$HitRatio_i = \sum_{u \in U} \frac{Hu,i}{|U|} \quad (4.2.1)$$

Likewise, average hit ratio is defined as the ratio of sum of the hit ratio for each item *i* following an attack on *i* across all items divided by the number of items:

$$\overline{HitRatio_i} = \sum_{u \in I} \frac{HitRatioi}{|I|} \quad (4.2.2)$$

5. Experimental Study

In our experiments, we have used News items as our dataset. The news items are read in the form of RSS feeds. Data in RSS feed is in standard XML format, hence semi structured. [Fig 5.1] shows the structure of the RSS file where data is organized in XML format.

News items are collected from various news channels like Times of India, Hindustan Times, IBN, CNN etc. in the form of RSS feeds. This data is then stored in MongoDB, a NOSQL database system [18]. Figure 5.2 depicts the structure of 'newsdata' collection of 'newsportal' database stored in Mongo database. When a user logs in to the system, he is presented with the news articles stored in the database. A logged in user submits his rating for all the viewed news items. These ratings are then used to predict the ratings for an unrated item in our recommender system. The work we presented here discusses the effect of average and random attack models on recommender system. We conducted our experiment on News Items database, 'newsdata'. The recommender system was subjected to average and random attack models for push attacks. For both the attack models, we inserted a number of attack profiles according to the attack size in the database and measured the prediction shift.

Figure 5.3 depicts the average prediction shift for average and random product push attack models against user-based collaborative recommender system for various attack sizes. From the results, we conclude that average attack has better prediction shift.

Figure 5.4 shows the average hit ratio for average and random product push attack models against user-based collaborative recommender system for various attack sizes. The average hit ratio of average attack is not indistinctively greater than that of the random attacks. Hence a remarkable prediction shift does not always guarantee higher hit ratio.

```
<rss version="2.0">
        <channel>
        <title>RSS feed for Times of India</title>
        <link>http://timesofindia.indiatimes.com</link>
        <description>Generated by NewsRack crawler</description>
        <pubDate>Wed, 21 Jan 2015 13:30:01 +0530</pubDate>

        <item>
        <title>
        Delhi man held with 600 sim cards in Muradnagar house
        </title>
        <link>
        http://timesofindia.indiatimes.com/city/noida/Delhi-man-held-with-600-sim-cards-in-Muradnagar house/articleshow/45960809.cms
        </link>
        <description>
        Delhi man held with 600 sim cards in Muradnagar house
        </description>
        <guid>
        http://timesofindia.indiatimes.com/city/noida/Delhi-man-held-with-600-sim-cards-in-Muradnagar-house/articleshow/45960809.cms
        </guid>
        </item>
        </channel>
</rss>
```

Figure 5.1 Structure of RSS file

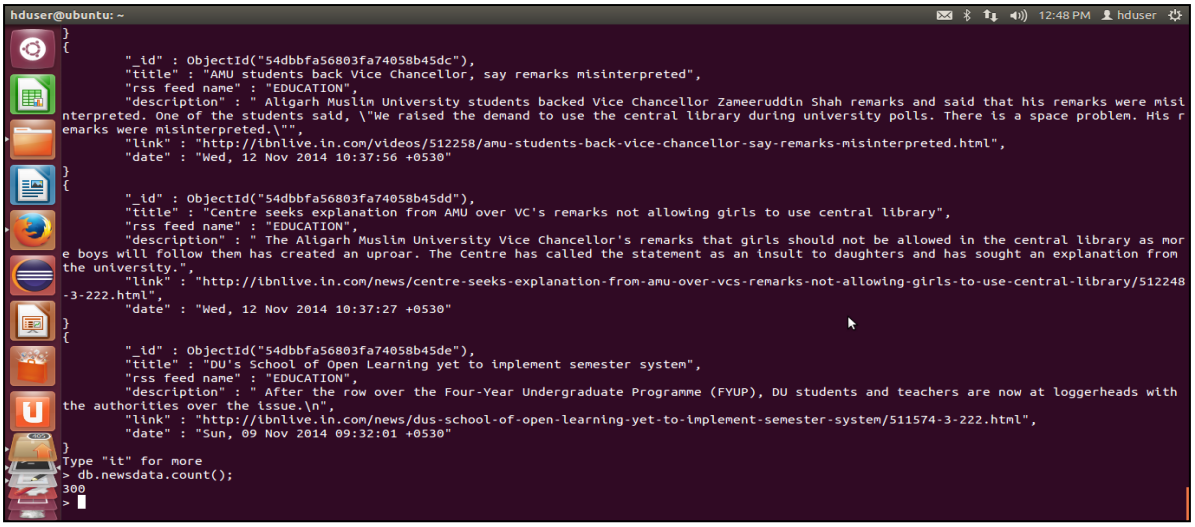

Figure 5.2 Structure of 'newsdata' collection within 'newsportal' database stored in MongoDB

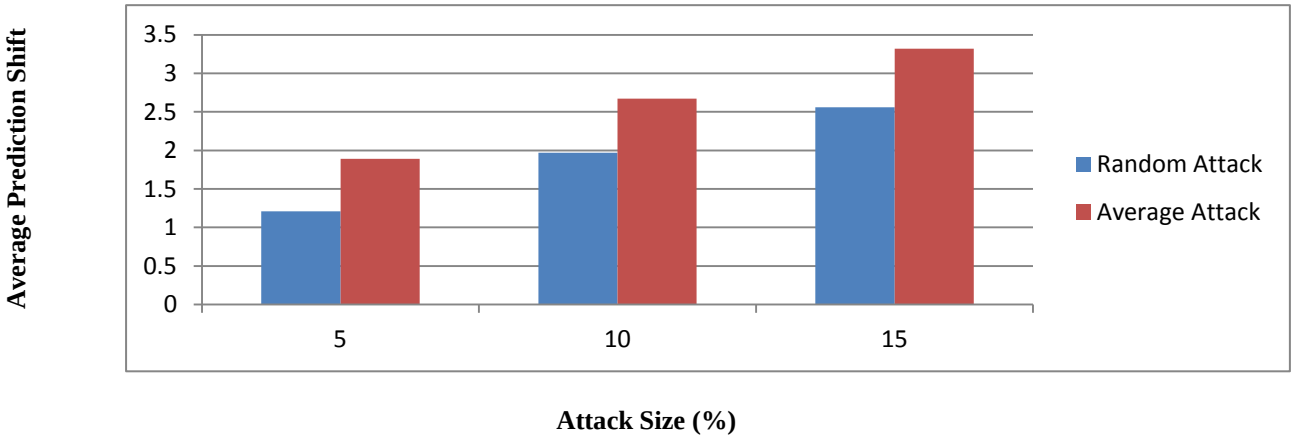

Figure 5.3 Average Prediction shift for average and random product push attacks

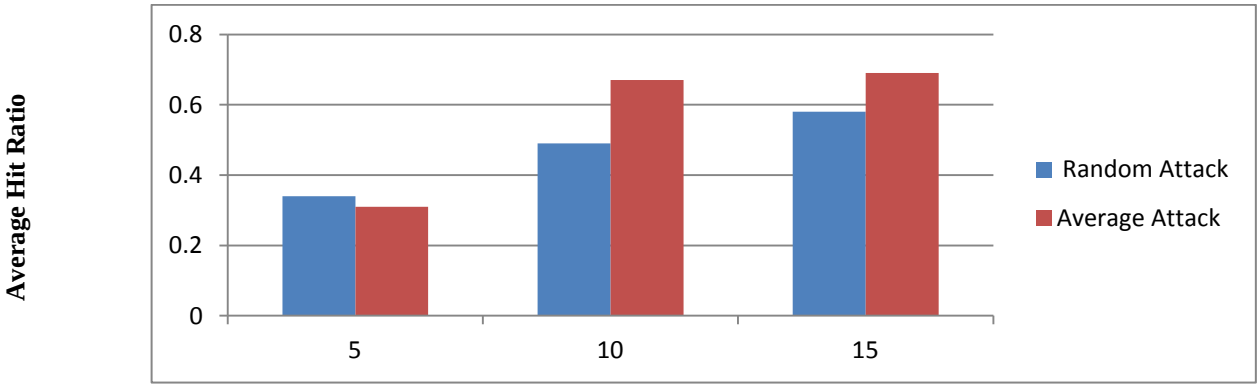

Figure 5.4 Average Hit Ratio for average and random product push attack

6. Conclusion

Recommender systems are one of the most important tools used these days to combat information overload. Now-a-days, recommender systems thrive on generating meaningful and useful recommendations by making use of user personal information. In this paper we have provided a comprehensive study of various attack models, methods used for the detection of fake user profiles, and number of approaches used by researchers to deal with these attacks. We also discussed the effectiveness of random and average attack models.

Our future work includes developing a low-computational cost secure recommender system capable of identifying various shilling attack models.

References

[1] Ackerman, M.S., Cranor, L.F., Reagle, J.: Privacy in e-commerce: Examining user scenarios and privacy preferences. In: Proceedings of 1st ACM Conference on Electronic Commerce (EC'99), pp. 1–8. New York, NY (1999)

[2] Lam S.K.., Frankowski D., Riedl J. (2006) Do You Trust Your Recommendations? An Exploration of Security and Privacy Issues in Recommender Systems. In: Müller G.(eds) Emerging Trends in Information and Communication Security. Lecture Notes in Computer Science, vol 3995. Springer, Berlin, Heidelberg.

[3] Lam, S.K., Riedl, J.: Shilling recommender systems for fun and profit. In Proceedings of the 13th International World Wide Web Conference pp. 393–402 (2004).

[4] Mobasher, B., Burke, R., Bhaumik, R., Williams, C.: Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. ACM Transactions on Internet Technology 7(4) (2007).

[5] Mobasher, B., Burke, R., Bhaumik, R., Williams, C.: Effective attack models for shilling item-based collaborative filtering system. In Proceedings of the 2005 WebKDD Workshop (KDD'2005) (2005).

[6] Chirita, P.A., Nejdl, W., Zamfir, C.: Preventing shilling attacks in online recommender systems. In Proceedings of the ACM Workshop on Web Information and Data Management (WIDM'2005) pp. 67–74 (2005).

[7] 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.

[8] J. R. Douceur, "The Sybil Attack", in Proceedings of the 1st International Workshop on Peer-to-Peer Systems, 2002.

[9] Zhang S, Ford J, Makedon F (2006) Deriving private information from randomly perturbed ratings. In: SDM. SIAM, pp 59–69.

[10] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for web transactions. ACM Tran. on Information and System Security, 1(1):66–92, 1998.

[11] H. Polat and W. Du. Privacy-preserving collaborative filtering using randomized perturbation techniques. In Proc. of the 3rd IEEE ICDM, pages 625–628, 2003.

[12] H. Polat and W. Du. SVD-based collaborative filtering with privacy. In Proc. of the 20th ACM Symp. on Applied Computing, pages 791–795, 2005.

[13] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. On the privacy preserving properties of random data perturbation techniques. In Proc. of the 3rd IEEE ICDM, pages 99–106, 2003.

[14] Z. Huang, W. Du, and B. Chen. Deriving private information from randomized data. In Proc. of the ACM SIGMOD, pages 37–48, 2005.

[15] Canny J (2002) Collaborative filtering with privacy. In: Security and privacy. Proceedings of the 2002 IEEE symposium on. IEEE, pp 45–57.

[16] Erkin, Zekeriya, et al. Privacy enhanced recommender system. IEEE Benelux Information Theory Chapter, 2010.

[17] Badsha, Shahriar, Xun Yi, and Ibrahim Khalil. "A practical privacy-preserving recommender system." Data Science and Engineering 1.3 (2016): 161-177.

[18] Gautam, Anjali, Tulika, Radhika Dhingra, and Punam Bedi. "Use of NoSQL Database for Handling Semi Structured Data: An Empirical Study of News RSS Feeds." Emerging Research in Computing, Information, Communication and Applications. Springer, New Delhi, 2015. 253-263.