CIT. Journal of Computing and Information Technology, Vol. 26, No. 1, March 2018, 45–56
doi: 10.20532/cit.2018.1003948

An Obfuscated Attack Detection Approach for Collaborative Recommender Systems

Saakshi Kapoor, Vishal Gupta and Rohit Kumar

Department of Computer Science and Engineering, UIET, Panjab University, Chandigarh, India

In the recent times, we have loads and loads of information available over the Internet. It has become very cumbersome to extract relevant information out of this huge amount of available information. So to avoid this problem, recommender systems came into play, which can predict outcomes according to user's interests. Although recommender systems are very effective and useful for users, the most used type of recommender system, i.e. collaborative filtering recommender system, suffers from shilling/profile injection attacks in which fake profiles are inserted into the database in order to bias its output. With this problem in mind,we propose an approach to detect attacks on recommender systems using Random Forest Classifier and find that, when tested at 10% attack, our approach outperformed earlier proposed approaches.

ACM CCS (2012) Classification: Computing Methodologies → Machine Learning → Machine Learning approaches → Classification and Regression Trees

Human-centered computing → Collaborative and Social Computing → Collaborative and Social Computing theory, concepts and paradigms → Collaborative filtering

Keywords: collaborative recommender systems, obfuscated attack, random forest classifier, SVM

1. Introduction

We are living in an era of information overload wherein information is being incessantly bombarded and people are breathlessly embracing the wide array of information available to them vis-à-vis varied sources. The backlash against the information overload is getting stronger than ever and is assuming gigantic proportions. Thus, one tool which has been developed to tackle such problems is recommender system [1].

Recommender systems [2], [5] can filter out the information required by the user from the vast amount of information available using certain characteristics and thus this concept is very helpful in overcoming the problem of information overload. Recommender systems can broadly be categorized as content-based [22], collaborative [4], [18], [23], [24] and Hybrid [3] recommender systems. In this paper, we will be focusing on collaborative recommender systems [2] because these are the ones which are widely used in today's era because of their abilities to give output to users according to their own needs. It recommends items based on similarity measures between users and items and recommends items to users that were liked by other users who have exhibited similar tastes. Thus it can easily handle even unusual requests by users which were not possible with content-based recommender systems. Although collaborative recommender systems are quite helpful in many ways, they are still prone to a shilling or profile injection attacks due to their natural openness. In these attacks, malicious users are inserted into existing dataset in order to influence the result of recommender systems. Mostly these attacks are generated by product sellers or developers who aim to promote their own product or demote their competitor's product.

Based on different assumptions attack models [28] can be divided into different categories such as push [27] or nuke [27] attacks and stan-

dard [4] or obfuscated [26] attacks. Although previous research is effective in detecting standard attacks, it is not as effective as it should be for obfuscated attacks.

Thus, how can we effectively identify and resist profile injection attacks [14], [29] has become an urgent problem that needs to be solved for the better development and extensive application of collaborative recommender systems.

In this paper, we propose a technique to detect profile injection attacks using Random Forest Classifier and conduct experiments on the MovieLens dataset [25] – 1M to verify the effectiveness of the proposed approach by comparing it with different classifier techniques. The paper is organized as follows: Section 2 describes the related work, in Section 3 we discuss details about our proposed approach, Section 4 deals with the experiments performed and their analysis and finally in Section 5 we conclude the paper along with the possible future scope.

2. Related Work

Shilling or Profile Injection Attacks pose a serious threat to recommender systems and collaborative filtering recommender systems which are highly vulnerable to these attacks. The term "shill" means posing as a fake user so as to decoy others. There have been researches regarding detection of these attacks and reducing their effects on recommender systems. Since most of the fake profiles appear similar to genuine profiles it is quite difficult to identify them. The main work carried out in this field falls into two categories: techniques to increase the robustness of recommender systems and techniques for detecting biased profiles, a technique which we will be focusing on in our work. In this section, we will be discussing some of the previous research work that has been carried out in the field of detecting shilling attacks.

Zhou *et al*. (2012) [7] proposed a hybrid unsupervised detection approach based on signal processing theory and designed a high pass filter to filter out the signal of attack profiles. The main advantage of this model was that it could detect attacks even if the number of attack profiles was unknown. Zhang *et al*. (2012) [8] proposed Meta-learning based approach to detect shilling attacks and also proposed an algorithm to create diverse base level training sets through a flexible combination of various attack types. They generated a Meta level classifier by combining various base level classifiers. Zhang *et al*. (2014) [9] proposed a spectral clustering method to make recommender systems resistant to shilling attacks in the cases when the attack profiles are highly correlated with each other. To estimate the highly correlated group they worked by first translating the matrix into a graph and then applying a spectral clustering algorithm to find the min-cut solution which will be used to estimate highly correlated group. Williams *et al*. (2008) [11] focused on Probabilistic Latent Semantic Analysis (PLSA) based technique that might be used to reduce the impact of segment attack and also examined their robustness against traditional attack models by using a model based on PLSA that aims to offer significant improvements in stability and robustness against all identified attack profiles. Zhou *et al*. (2014) [12] proposed and evaluated an algorithm for detecting average and random attacks first by using a rough detection model that will be used to divide the profiles into genuine and potential attack profiles. Then the attack profiles set will be analysed, based on which genuine profiles will be further removed from the target item set. Yang *et al*. (2015) [13] proposed an unsupervised method to detect shilling attacks first by filtering out genuine users, using suspected target items as far as possible so as to reduce time consumption. Then they developed a new similarity metric by combining traditional similarity metric and linkage information between users to improve accuracy of similarity of users. Q. Zhou (2016) [15] proposed a supervised approach to detect obfuscated attacks, especially Average over Popular (AoP) attack by using the theory of term frequency-inverse document frequency (TFIDF) to extract the features of attack and then used SVM (Support Vector Machine) to generate a SVM based classifier. Finally, they used the generated classifier to detect AoP attack. Zhang *et al*. (2014(a)) [16] proposed an ensemble detection model by introducing Back Propogation (BP) neural network and ensemble learning technique. They created Ensemble Detection Model (EDM) through the combination of various attack types, created base training sets which included samples of attack profiles and

had great diversities with each other. Then they created base training sets to train BP neural networks to generate diverse base classifiers and finally selected parts of base classifiers which had the highest precision on validation dataset and integrated them using voting strategy. Zhang *et al*. (2014(b)) [17] proposed a method to detect profile injection attacks by combining Hilbert Huang Transform (HHT) and SVM and making them work incrementally. They worked by constructing rating series for each user profile based on novelty and popularity of items. Then they used Empirical Mode Decomposition (EMD) to decompose each rating series and extract Hilbert spectrum based features for characterizing profile injection attacks. Finally, they used SVM to detect profile injection attacks based on the proposed features. As discussed above in [8], Bhebe *et al*. (2015) [19] also proposed a combiner strategy that combined multiple classifiers to detect shilling attacks in which KNN, SVM and Bayesian networks acted as initial base classifiers and Naïve Bayes was used as meta-classifier. Chung *et al*. (2013) [21] proposed Beta Protection (βP) to alleviate recommender systems and make them resistant to shilling attacks. βP was based on βeta distribution to detect and remove fake user profiles.

Sutter *et al*. (2008) [34] proposed a generic method to utilize tags as a supplementary source to predict item recommendations. Verbert *et al*. (2011) [35] presented datasets that can capture learner interactions with tools and resources and can be used for learning analytics research. Manouselis *et al*. (2011) [36] discussed how Technology Enhanced Learning (TEL) can design, test and develop socio-economic innovations so as to practice them in real life scenarios as well. Konstan *et al*. (2012) [37] presented a review about how recommender algorithms can be applied to real life scenarios so as to predict better recommendations.

Above, we discussed different research work that has been carried out in the field of attack detection, but how a biased user can affect results of a recommender system can be explained with the help of an example which is presented below.

Consider, for example, a recommender system that identifies movies a user might like to watch, using a user-based collaborative algorithm. A user profile in this hypothetical system consists of the user's ratings of various movies on a scale of 1–5 with 1 being the lowest. Table 1 shows Eden's profile (genuine user) along with that of four genuine users. An attacker, Eve, has inserted attack profiles into the system. Eve's attack profiles may be closely related to the profiles of one or more existing users or they may be based on average or expected ratings of items across all users.

Suppose the system is using a simplified user-based collaborative filtering approach where the predicted ratings for Eden on target will be obtained by finding the closest neighbour to Eden based on Pearson similarity. Without the attack profile the most similar user to Eden would be User 2, but after the attack the most similar user to Eden will be Eve. So in this example the attack will be successful and Eden will get a recommendation according to the attack and not according to a genuine user. Thus we can clearly see that even a single attack profile can affect the result of recommender system to a great extent and that these attacks can be identified with the help of some detection attributes.

Here, if Eve was not added as an authentic user, then User 2 would have been similar to Eden, but after the addition of attacker Eve, Eve is going to be most similar to Eden.

Thus it shows that how addition of an attacker can affect the results.

Here, the Pearson similarity is calculated as shown in the expression below. Similarly, r (Eden, Eve) = 0.72.

$$r(\text{Eden}, \text{User 1}) = \frac{\left(\frac{7}{4}\right)\left(-\frac{1}{5}\right)+\left(-\frac{1}{4}\right)\left(-\frac{11}{5}\right)+\left(\frac{3}{4}\right)\left(-\frac{6}{5}\right)+\left(-\frac{9}{4}\right)\left(\frac{9}{5}\right)}{\sqrt{\left(\frac{7}{4}\right)^2+\left(-\frac{1}{4}\right)^2+\left(\frac{3}{4}\right)^2+\left(-\frac{9}{4}\right)^2}\sqrt{\left(-\frac{1}{5}\right)^2+\left(-\frac{11}{5}\right)^2+\left(-\frac{6}{5}\right)^2+\left(\frac{9}{5}\right)^2+\left(\frac{9}{5}\right)^2}} = -0.45$$

Table 1. An example of how an attack can affect the results.

	Movie 1	Movie 2	Movie 3	Movie 4	Target	Pearson Similarity [1]
Eden	5	3	4	1	?	
User 1	3	1	2	5	5	–0.45
User 2	4	3	3	3	2	0.41
User 3	3	3	1	5	4	–0.68
User 4	1	5	5	2	1	–0.02
Eve	5	3	4	3	5	0.72

Table 2. Modification of Table 1 by taking an average of rows.

	Movie 1	Movie 2	Movie 3	Movie 4	Target	Average
Eden	5	3	4	1	?	13/4
User 1	3	1	2	5	5	16/5
User 2	4	3	3	3	2	3
User 3	3	3	1	5	4	16/5
User 4	1	5	5	2	1	14/5
Eve	5	3	4	3	5	4

Table 3. Modification of Table 3 by subtracting an average from each rating element.

	Movie 1	Movie 2	Movie 3	Movie 4	Target
Eden	7/4	–1/4	–3/4	–9/4	
User 1	–1/5	–11/5	–6/5	9/5	9/5
User 2	1	0	0	0	–1
User 3	–1/5	–1/5	–11/5	9/5	4/5
User 4	–9/5	11/5	11/5	–4/5	–9/5
Eve	1	–1	0	–1	1

3. Proposed Detection Method

The proposed detection model is based on the idea that a user might be an attacker and might be wishing to alter the results of the recommender system. The basic goal of our approach is the identification of such users. Figure 1 describes the framework of our proposed detection approach for detecting attacks. Our framework consists of three stages.

In the first stage, certain attributes are computed on ratings assigned by users in the dataset. Next, training and testing sets are used to generate Classifier for the purpose of shilling attack detection. In the final stage of detection, the proposed Classifier is used to detect the results.

3.1. Phase 1: Attribute Extraction

Attribute extraction forms a key factor in determining the performance of shilling attack detection. Some of the attributes which will be used are RDMA (Rating Deviation from Mean Agreement), WDA (Weighted Degree of Agreement), Similarity (cosine and Pearson similarity), LenVar (Length Variance), and along with that, we will be using TF-IDF (Term Frequency-Inverse Document Frequency) [15] as well as a few combinations of the said attributes. With the help of a matrix example (Table 4), we will explain how these attributes are calculated.

1. Rating Deviation from Mean Agreement (RDMA): RDMA [19] can identify attackers by analysing the profile's average deviation per

Table 4. Ratings assigned for a movie by a user.

User or Movie	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5
User A	5	4	3	2	1
User B	1	3	2	4	1
User C	1	2	2	3	5
User D	2	3	5	4	1
User E	1	4	5	2	3

item or user. It is defined as:

$$\mathrm{RDMA}_x = \frac{\sum_{x=0}^{T_u} \frac{\left|r_{x,i} - \overline{r_i}\right|}{R_{x,i}}}{N_x}$$

where T_u is the number of items user x rated, $r_{x,i}$ is the rating given by the user x to item i, r_i is the average rating of item i, $R_{x,i}$ is the number of ratings provided for item i by all users and N_x is the number of users. For instance, the RDMA value for User A is computed as shown below:

$$\mathrm{RDMA}_{\text{User A}} = \frac{\left(\frac{|5-2|}{5}\right)}{5} + \frac{\left(\frac{|4-3.2|}{5}\right)}{5} + \frac{\left(\frac{|3-3.4|}{5}\right)}{5} + \frac{\left(\frac{|2-3|}{5}\right)}{5} + \frac{\left(\frac{|1-2.2|}{5}\right)}{5}$$

Here, for the example, we have taken $R_{x,i} = 5$.

2. Weighted Degree of Agreement (WDA): WDA [6] can be calculated as the numerator of RDMA.

$$\mathrm{WDA}_x = \sum_{x=0}^{T_u} \frac{\left|r_{x,i} - \overline{r_i}\right|}{R_{x,i}}$$

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

$$\mathrm{WDA}_{\text{User B}} = \frac{|1-2|}{5} + \frac{|3-3.2|}{5} + \frac{|2-3.4|}{5} + \frac{|4-3|}{5} + \frac{|1-2.2|}{5}$$

Here, for the example, we have taken $R_{x,i} = 5$

3. Length Variance (LengthVar): LengthVar [6] is used to capture how much the length of a given profile varies from the average length in the dataset. It is particularly effective in detecting attacks with large filler sizes.

$$\mathrm{LenghtVar} = \frac{\left|\#score_j - \overline{\#score}\right|}{\sum_{i=0}^{N}\left(\#score_i - \overline{\#score}\right)^2}$$

where $\#score_j$ is the average length of a profile in the rating database and N is the total number of users in the system.

$$\mathrm{LenghtVar}_{(for\ a\ \text{User})} = \frac{5-3}{(5-3)^2 + (5-3)^2 + (5-3)^2 + (5-3)^2 + (5-3)^2}$$

Here, for the example, we have taken the average profile length of the user as 3.

4. The Degree of Similarity with Top Neighbours (DegSim): DegSim [12] is used to capture the average similarity of a profile's k nearest neighbours

$$\mathrm{DegSim} = \sum_{i=1}^{x} Z_{i,j}$$

where $Z_{i,j}$ is the Pearson correlation between users i and j, and x is the number of neighbours. $Z_{1,2} = 0.122$ (Pearson correlation as discussed in Section 2)

$$\mathrm{DegSim} = 0.122$$

As we have calculated DegSim for User 1, therefore, number of neighbours = 1.

5. To use the theory of TF-IDF [15], to extract the features for an attack, we first converted the rating dataset to a 0, 1 format by converting rat-

ings. Numbers here denote the value of the ratings and ∅ denotes the user that does not rate the item. Here, Table 5 acts as a rating database and in Table 6, ratings are converted to 1 and ∅ is converted to 0. We used TF-IDF as described below:

$$\text{TFIDF}(u, m) = \text{TF}(u,m) \times \text{IDF}(m)$$

TF $(u, m) = 0$ if rating = 0 else TF (u, m) = 1/number of ratings rated by a user. IDF (m) = log (genuine users/n_u), where n_u is the number of users who have not given rating 0 to movie m and belong to genuine users. $TF(u, m)$ is the term frequency for user u and movie m and IDF (m) is the inverse document frequency for movie m.

$$\text{TF}\left(\text{User A, Movie 1}\right) = \frac{1}{5}$$

$$\text{IDF}\left(\text{Movie 1}\right) = \log\left(\frac{5}{4}\right)$$

$$\text{TF-IDF}\left(\text{User A, Movie 1}\right) = \frac{1}{5} * \log\left(\frac{5}{4}\right) =$$

$$= \frac{1}{5} \times 0.22314$$

$$= 0.044628$$

3.2. Phase 2: Shilling Detection Algorithm

We propose an approach for the detection of shilling attacks by examining various classifier models like SVM [20], [31], Random forest [32] and MLP (multilayer perceptron) [33], etc. and then select the classifier model which outperform other models on the basis of certain evaluation metrics like precision and recall. The proposed attack detection algorithm is described below:

Algorithm 1. The proposed attack detection.

Input: ratings
Output: result
For i = 0 to length (ratings) do
  Calculate selected attributes i.e. RDMA, WDA, Length Variance, etc.
end for
For all attributes ε selected attributes do
  Classifier ← train using one of the selected attributes and repeat for all other attributes
 result ← ∅
 result ← result ∅ {classifier detection result}
end for
Return result

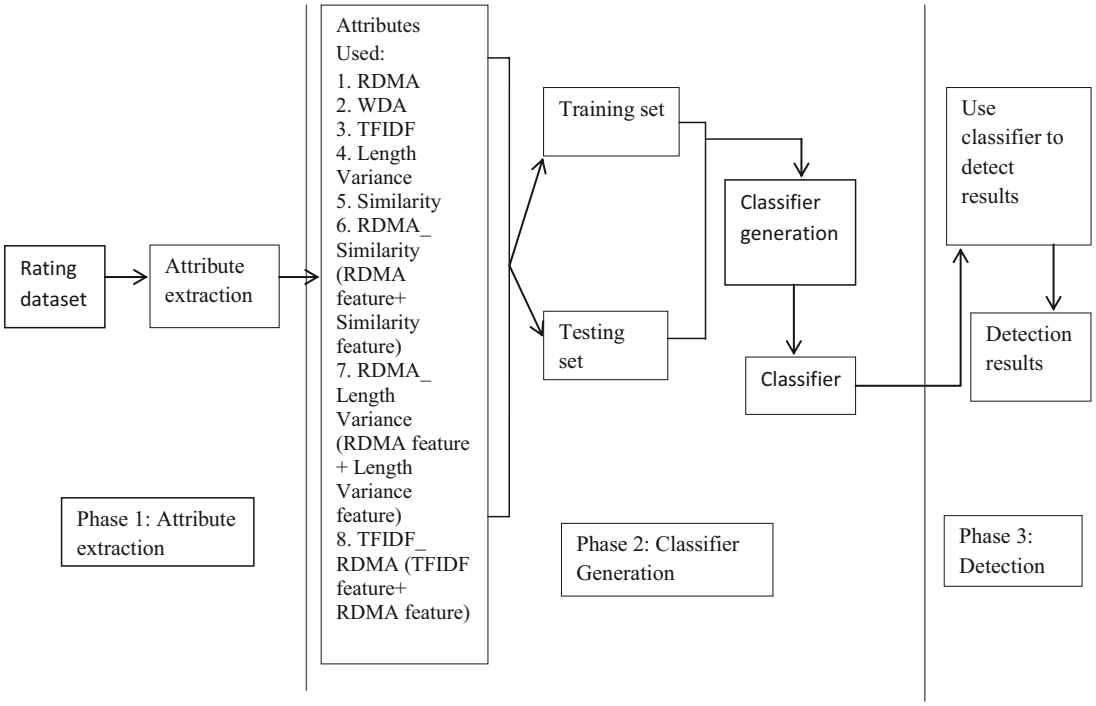

Figure 1. Framework of the proposed approach.

Table 5. Example of rating dataset.

User or Movie	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5
User A	5	4	3	2	1
User B	1	3	2	4	1
User C	1	2	Ø	3	5
User D	2	3	5	4	1
User E	1	4	5	2	3

Table 6. Converting table 7 for TF-IDF calculation.

User or Movie	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5
User A	1	1	1	1	1
User B	1	1	1	1	1
User C	1	1	0	1	1
User D	1	1	1	1	1
User E	1	1	1	1	1

After various experiments, it was found out that Random Forest Classifier outperforms other classifiers while taking care of different attack types and various attributes. It performed much better than other classifiers except in a few cases. Random Forests are basically based on the concept of Decision Trees but here instead of just using a single Decision Tree, Random Forests take up many Decision Trees. For example, for our approach we have taken a Decision Tree count of 100. Furthermore, as discussed above, Random Forest is based on the basic structural principle of Decision Trees, but it works by constructing a multitude of Decision Trees at training time and outputting the class that is the mean or mode of the individual trees. In this way, it forms an ensemble learning method. It can easily classify large amounts of data with accuracy.

3.3. Phase 3: Detection

We used the generated classifier to detect results for shilling attack by using certain evaluation metrics such as precision and recall.

4. Experiments and Evaluation

In this section, we present the overall performance comparison of our approach with some other techniques and classifiers.

4.1. Dataset

In our experiments, we have used the publicly available MovieLens 1M dataset [25]. This dataset consists of 1,000,209 ratings by 6040 MovieLens users of approximately 3900 movies, which contains all genuine users. All ratings were integer values between 1 and 5, 1 being the lowest (disliked) and 5 the highest (liked). For our experiments, we chose 100 random users from the dataset.

For non-genuine users, we designed a dataset containing only malicious users. We used different attack models to create different attack profiles. Thus we designed attack dataset (containing attack ratings for different malicious users) for different attacks according to the said attacks. For AoP attack [15], [26], [30], the set of attacked items consists of movies which fall into the category of most 5% rated movies by users. For User Shifting attack [26], [30], we selected some of the ratings from each profile and lowered their ratings by one. For Noise Injection attack [26], [30], we were required to calculate standard normal distribution on the dataset and multiply that by ratings and that came out to be 0.4. To create the test set, attack profiles were injected individually into the dataset. In this paper, we only detect push attacks but this approach will be applicable to nuke attacks as well.

4.2. Evaluation Metrics

In the context of attack detection, our goal was to provide insight into how accurately the algorithm identifies attack profiles. For measuring detection performance, we used precision [17], recall [10] as our evaluation metrics which are defined below:

$$\text{Precision} = \frac{\text{Correctly identified attackers}}{\text{Correctly identified attackers} + \text{Wrong identified attackers}}$$

$$\text{Recall} = \frac{\text{True identified attackers}}{\text{True identified attackers} + \text{Missed attackers}}$$

4.3. Experimental Results and Analysis

4.3.1. Comparison of Recall and Precision

We conducted several experiments and compared the performance of the proposed algorithm with different classifiers as well as with different existing techniques. Tables 7, 8, 9 and figures 2, 3, 4 give a comparative performance of different classifiers we used for detecting the best classifier amongst them all for our problem and Table 10 and Figure 2 give a comparison between different techniques available for AoP attack detection on the basis of precision and recall.

4.3.2. Complexity of the Proposed Method

The average space complexity of the proposed algorithm is 0 (5n).

5. Conclusion and Future Work

The issue of Shilling attacks is a major concern in the field of recommender systems. To maintain its trustworthiness, we need to either design recommender systems in such a way that they are resistant to such attacks or design algorithms which can detect attacks easily and

Table 7. Precision comparison of different classifiers at 10% AoP Attack.

Attributes\Classifiers	SVM	Naïve Bayes	KNN	Random Forest	MLP
1. RDMA	0.47	0.275	0.35	0.40	0.325
2. WDA	0.80	0.70	0.85	0.98	0.85
3. Similarity	0.92	0.94	0.95	0.99	0.97
4. Length Variance	0.96	0.99	0.90	0.97	0.99
5. TFIDF	0.98	0.94	0.96	0.99	0.98
6. RDMA + Similarity	0.14	0.6375	0.6625	0.65	0.6375

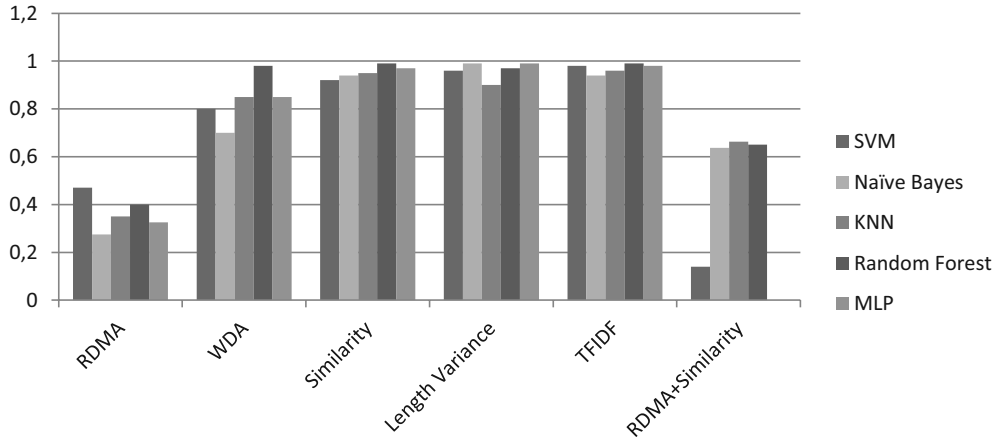

Figure 2. Precision comparison of different classifiers at 10% AoP Attack.

Table 8. Precision comparison of different classifiers at 10% Noise Injection Attack.

Attributes\Classifiers	SVM	Naïve Bayes	KNN	Random Forest	MLP
1. RDMA	0.40	0.475	0.425	0.525	0.525
2. WDA	0.70	0.70	0.375	0.99	0.99
3. Similarity	0.99	0.99	0.98	0.98	0.94
4. Length Variance	0.98	0.99	0.99	0.99	0.98
5. TFIDF	0.96	0.94	0.89	0.98	0.94
6. RDMA+Similarity	0.16	0.70	0.7	0.7625	0.70

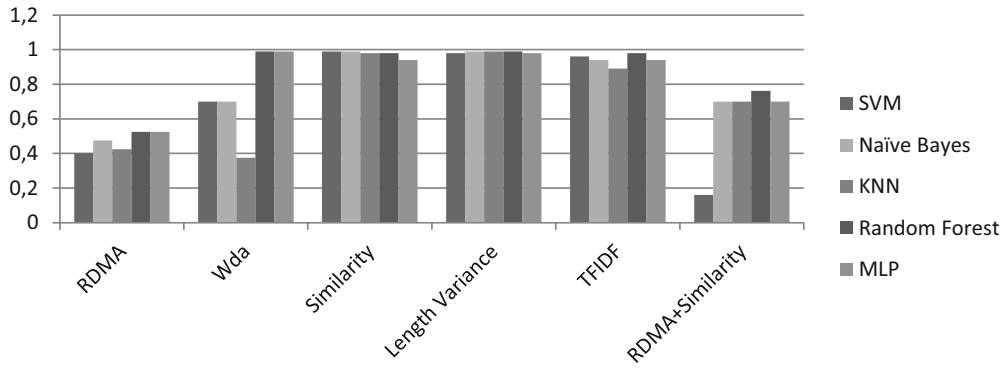

Figure 3. Precision comparison of different classifiers at 10% Noise Injection Attack.

Table 9. Precision comparison of different classifiers at 10% User Shifting Attack.

Attributes\Classifiers	SVM	Naïve Bayes	KNN	Random Forest	MLP
1. RDMA	0.23	0.275	0.35	0.35	0.325
2. WDA	0.70	0.85	0.85	0.96	0.93
3. Similarity	0.97	0.94	0.94	0.98	0.97
4. Length Variance	0.99	0.97	0.96	0.99	0.98
5. TFIDF	0.96	0.96	0.96	0.99	0.98
6. RDMA+Similarity	0.21	0.6375	0.6625	0.65	0.6375

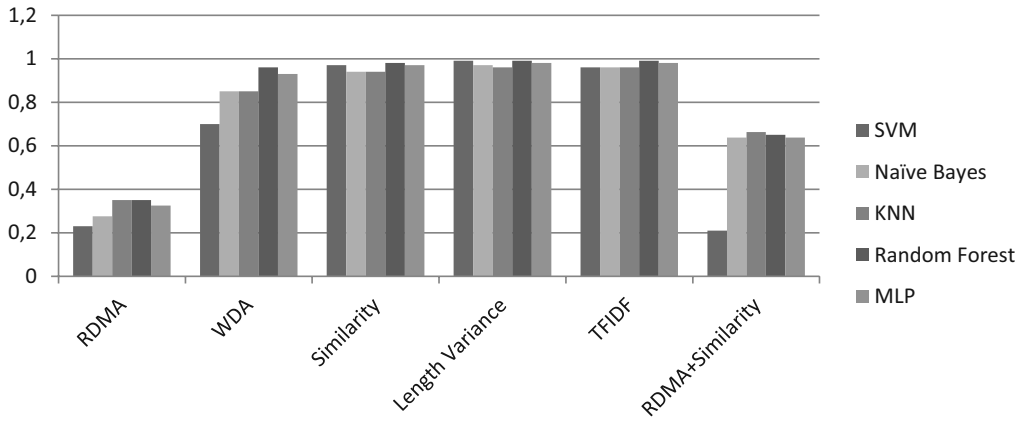

Figure 4. Precision comparison of different classifiers at 10% User Shifting Attack.

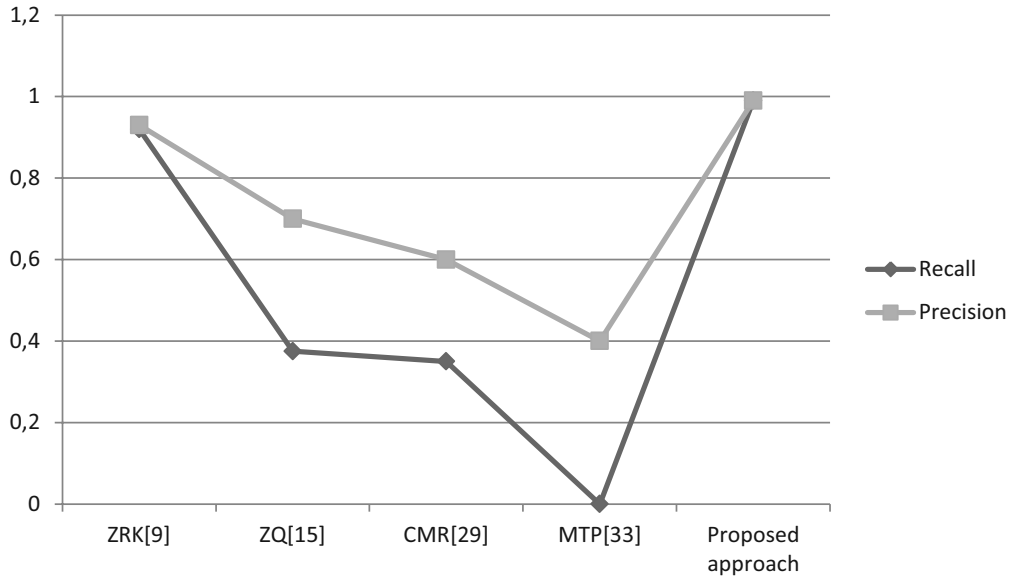

Figure 5. Recall and Precision comparison of different available techniques at 10% AoP Attack.

Table 10. Comparison of different techniques at 10% AoP attack.

Technique	Recall	Precision
1. Zhang *et al*. (2014) [9]	0.92	0.93
2. Q.Zhou (2016) [15]	0.375	0.7
3. Williams *et al*. (2007) [29]	0.35	0.6
4. Mehta *et al*. (2007) [33]	–	0.4
5. Proposed (with Random Forest)	0.9825	0.9825

effectively. Our proposed approach is carried out for this purpose only i.e. designing of an algorithm for easy and effective detection of attacks. For this purpose, our paper demonstrates an attack detection model based on Random Forest Classifier and we conducted several experiments on 1M MovieLens dataset. In our future work, we intend to extend and improve attack detection by

1. Working on more types of attacks i.e. standard and obfuscated attacks.
2. Working on different features and attributes for attack detection.
3. Trying to improve results and reduce computation time by using most influential users instead of the whole dataset.

Although our proposed approach is quite effective, it will be hard to see if it can perform successfully in real life scenarios as well or not.

References

[1] A. Dhoha *et al*., "A Survey Paper on Recommender Systems", arXiv preprint: 1006.5278, 2010.

[2] P. Melville and V. Sindhwani, "Recommender Systems", in Encyclopedia of Machine Learning, pp. 829–838, Springer US, 2011. http://dx.doi.org/10.1007/978-1-4899-7687-1_964

[3] R. Burke, "Hybrid Recommender Systems: Survey and Experiments", *User Modeling and User-adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[4] B. Mobasher *et al*., "Attacks and Remedies in Collaborative Recommendation", *IEEE Intelligent Systems*, vol. 22, no. 3, 2007. http://dx.doi.org/10.1109/MIS.2007.45

[5] F. O. Isinkaye *et al*., "Recommendation Systems: Principles, Methods and Evaluation", *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261–273, 2015. http://dx.doi.org/10.1016/j.eij.2015.06.005

[6] Z. Yang, and Z. Cai, "Detecting Abnormal Profiles in Collaborative Filtering Recommender Systems", *Journal of Intelligent Information Systems*, pp. 1–20, 2016. http://dx.doi.org/10.1007/s10844-016-0424-5

[7] Q. Zhou and F. Zhang, "A Hybrid Unsupervised Approach for Detecting Profile Injection Attacks

in Collaborative Recommender Systems", *Journal of Information and Computational Science*, vol. 9, no. 3, pp. 687–694, 2012.

[8] F. Zhang and Q. Zhou, "A Meta-Learning-Based Approach for Detecting Profile Injection Attacks in Collaborative Recommender Systems," *Journal of Computers*, vol. 7, no. 1, pp. 226–234, 2012.
http://dx.doi.org/10.4304/jcp.7.1.226-234

[9] Z. Zhang and S. R. Kulkarni, "Detection of Shilling Attacks in Recommender Systems via Spectral Clustering", in *Information Fusion (FUSION), 17th International Conference on Information Fusion*, *IEEE*, 2014, pp. 1–8.

[10] Z. Zhang and S. R. Kulkarni, "Graph-based Detection of Shilling Attacks in Recommender Systems", in *Machine Learning for Signal Processing (MLSP)*, *International Workshop on Machine Learning for Signal Processing*, *IEEE*, 2013, pp. 1–6.
http://dx.doi.org/10.1109/MLSP.2013.6661953

[11] C. Williams *et al*., "Evaluation of Profile Injection Attacks in Collaborative Recommender Systems", Technical report. Available from:
http://facweb.cti.depaul.edu/research/techreports/TR06-006.pdf

[12] W. Zhou *et al*., "Attack Detection in Recommender Systems based on Target Item Analysis", in *Neural Networks (IJCNN)*, *International Joint Conference on Neural Networks*, *IEEE*, 2014, pp. 332–339.
http://dx.doi.org/10.1109/IJCNN.2014.6889419

[13] Z. Yang, "Defending Suspected Users by Exploiting Specific Distance Metric in Collaborative Filtering Recommender Systems", in *Data Mining Workshop (ICDMW)*, *International Conference on Data Mining Workshops*, *IEEE*, 2015, pp. 1001–1006.
http://dx.doi.org/10.1109/ICDMW.2015.89

[14] P. Chakraborty and S. Karforma, "Detection of Profile-injection Attacks in Recommender Systems Using Outlier Analysis", *International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA)*, *Procedia Technology*, vol. 10, pp. 963–969, 2013.
http://dx.doi.org/10.1016/j.protcy.2013.12.444

[15] Q. Zhou, "Supervised Approach for Detecting Average over Popular Items Attack in Collaborative Recommender Systems", *IET Information Security*, vol. 10, no. 3, pp. 134–141, 2016.
http://dx.doi.org/10.1049/iet-ifs.2015.0067

[16] F. Zhang and Q. Zhou, "Ensemble Detection Model for Profile Injection Attacks in Collaborative Recommender Systems based on BP Neural Network", *IET Information Security*, vol. 9, no. 1, pp. 24–31, 2014.
http://dx.doi.org/10.1049/iet-ifs.2013.0145

[17] F. Zhang and Q. Zhou, "HHT–SVM: An Online Method for Detecting Profile Injection Attacks in Collaborative Recommender Systems", *Knowledge-based Systems*, vol. 65, pp. 96-105, 2014.

[18] Z.-J. Deng *et al*., "Shilling Attack Detection in Collaborative Filtering Recommender System by PCA Detection and Perturbation", in *Wavelet Analysis and Pattern Recognition (ICWAPR)*, *International Conference on Wavelet Analysis and Pattern Recognition*, IEEE, 2016, pp. 213–218.
http://dx.doi.org/10.1109/ICWAPR.2016.7731644

[19] W. Bhebe and O. P. Kogeda, "Shilling Attack Detection in Collaborative Recommender Systems using a Meta Learning strategy", in *Emerging Trends in Networks and Computer Communications (ETNCC)*, *International Conference on Emerging Trends in Networks and Computer Communications*, *IEEE*, 2015, pp. 56–61.
http://dx.doi.org/10.1109/ETNCC.2015.7184808

[20] W. Zhou *et al*., "SVM-TIA a Shilling Attack Detection Method based on SVM and Target Item Analysis in Recommender Systems", *Neurocomputing*, vol. 210, pp. 197–205, 2016.
http://dx.doi.org/10.1016/j.neucom.2015.12.137

[21] C.-Y. Chung *et al*., "βP: A novel Approach to Filter Out Malicious Rating Profiles from Recommender Systems", *Decision Support Systems*, vol. 55, no. 1, pp. 314–325, 2013.
http://dx.doi.org/10.1016/j.dss.2013.01.020

[22] G. Adomavicius and A. Tuzhilin, "Toward the next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions", *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.
http://dx.doi.org/10.1109/TKDE.2005.99

[23] D. Almazro *et al*., "A Survey Paper on Recommender Systems", arXiv preprint: 1006.5278, 2010.

[24] X. Su and T. M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques", *Advances in Artificial Intelligence*, pp. 4, 2009.
http://dx.doi.org/10.1155/2009/421425

[25] F. M. Harper and J. A. Konstan, "The Movielens Datasets: History and Context", *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 5, no. 4, pp. 19, 2016.
http://dx.doi.org/10.1145/2827872

[26] R. Bhaumik *et al*., "A Clustering Approach to Unsupervised Attack Detection in Collaborative Recommender Systems", in *Proceedings of the 7th IEEE International Conference on Data Mining*, 2011, Las Vegas, NV, USA, pp. 181–187.

[27] F. Zhang, "A Survey of Shilling Attacks in Collaborative Filtering Recommender Systems", in *Computational Intelligence and Software Engineering*, *International Conference on Computational Intelligence and Software Engineering*, *IEEE*, 2009, pp. 1–4.
http://dx.doi.org/10.1109/CISE.2009.5365077

[28] I. Gunes *et al*., "Shilling Attacks against Recommender Systems: a Comprehensive Survey", *Artificial Intelligence Review,* pp. 1–33, 2014. http://dx.doi.org/10.1007/s10462-012-9364-9

[29] C. A. Williams *et al*., "Defending Recommender Systems: Detection of Profile Injection Attacks", *Service Oriented Computing and Applications*, vol. 1, no. 3, pp. 157–170, 2007. http://dx.doi.org/10.1007/s11761-007-0013-0

[30] C.-W. Hsu *et al*., "A Practical Guide to Support Vector Classification", pp. 1–16, 2003.

[31] L. Breiman, "Random forests", *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[32] D. W. Ruck *et al*., "Feature Selection using a Multilayer Perception", *Journal of Neural Network Computing*, vol. 2, no. 2, pp. 40–48, 1990.

[33] B. Mehta *et al*., "Lies and Propaganda: Detecting Spam users in Collaborative Filtering", in *Proceedings of the 12th International Conference on Intelligent user Interfaces*, *ACM*, 2007, pp. 14–21. http://dx.doi.org/10.1145/1216295.1216307

[34] T. Sutter *et al*., "Tag-Aware Recommender Systems by Fusion of Collaborative Filtering Algorithms", in *Proceedings of the 2008 ACM Symposium on Applied Computing*, *ACM*, 2008, pp. 1995–1999.

[35] K. Verbert *et al*., "Dataset-Driven Research for Improving Recommender Systems for Learning", in *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, *ACM*, 2011, pp. 44–53.

[36] N. Manouselis *et al*., "Recommender Systems in Technology Enhanced Learning", in Recommender Systems Handbook, Springer, Boston, MA, 2011, pp. 387–415.

[37] J. A. Konstan and J. Riedl, "Recommender Systems: from Algorithms to user Experience", *User Modeling and User-Adapted Interaction*, vol. 22, pp. 101–123, 2012. http://dx.doi.org/10.1007/s11257-011-9112-x

Received: December 2017
Revised: May 2018
Accepted: June 2018

Contact addresses:
Saakshi Kapoor
Department of Computer Science and Engineering
UIET, Panjab University
Chandigarh, India
e-mail: saakshikpr28@gmail.com

Vishal Gupta
Department of Computer Science and Engineering
UIET, Panjab University
Chandigarh, India
e-mail: vishal_gupta100@yahoo.co.in

Rohit Kumar
Department of Computer Science and Engineering
UIET, Panjab University
Chandigarh, India
e-mail: rklachotra@gmail.com

Saakshi Kapoor is currently pursuing her master's thesis under the guidance of Dr. Vishal Gupta and Mr. Rohit Kumar from UIET, Panjab University, Chandigarh, India. She has completed her BE in Computer Science from Chitkara University, Baddi Campus, Himachal Pradesh, India.

Vishal Gupta is currently working as a Senior Assistant Professor at UIET, Panjab University, Chandigarh, India. He joined UIET in 2006. He has completed his BE, ME and PhD degrees in Computer Science and Engineering. He received the Young Scientist Award in Engineering & Technology from Punjab Academy of Sciences for 2013. He has published a number of research papers. His major research interest lies in the areas of natural language processing and text mining.

Rohit Kumar is currently working as an Assistant Professor at UIET, Panjab University, Chandigarh, India. He joined UIET in 2011. He holds the BE and ME degrees and has published a number of research papers. His major research interests are in software engineering and cloud computing.