ICIC Express Letters
Volume **10**, Number **1**, January **2016**

 ISSN 1881-803X
pp. **59–66**

FEATURE EXTRACTION APPROACH FOR RECOMMENDATION ATTACKS BASED ON MUTUAL INFORMATION

Quanqiang Zhou

Computer Engineering Institute
Qingdao Technological University
No. 777, Jialingjiang Rd., Huangdao Dist., Qingdao 266520, P. R. China
zhouqiang128@126.com

Received July 2015; accepted October 2015

Abstract. *The features' capability of characterizing recommendation attacks is one key factor to determine the detection performance of the supervised approaches. The existing features are extracted mainly from the perspective of rating values. However, these features have limited capability of characterizing the recommendation attacks. With this problem in mind, in this paper from the perspective of rating distribution we propose a feature extraction approach for the detection of recommendation attacks based on the theory of mutual information. Firstly, according to the attributes of items, i.e., popularity and novelty, we divide the items into different sets. After that, we use the theory of mutual information to calculate the correlation between rated items and the above sets as features of characterizing recommendation attacks. The experimental results on MovieLens dataset show that the proposed features can well characterize the recommendation attacks.*

Keywords: Collaborative filtering, Recommendation attacks, Attack detection, Mutual information

1. **Introduction.** Collaborative filtering recommender systems [1] have been shown to have significant vulnerabilities to "shilling attacks" or called "recommendation attacks" [2, 3]. In order to detect such attacks, some unsupervised or supervised approaches have been proposed. Unsupervised approaches require certain prior knowledge [4, 5]. One advantage of unsupervised approaches is that they do not need labeled training samples. However, some prior knowledge used in these approaches is difficult to get in real world. Supervised approaches need training samples [6, 7, 8, 9]. In the stage of feature extraction, most of the supervised approaches employ features proposed by Williams et al. [6, 7] (called Williams' features). Williams' features are extracted mainly from the perspective of rating values. One key factor to determine the detection performance of the supervised approaches is the features' capability of characterizing the recommendation attacks. However, Williams' features have limited capability of characterizing these attacks.

To improve the features' capability of characterizing the recommendation attacks, in this paper we propose a feature extraction approach based on mutual information (MI). The main contributions are described as follows. (1) We propose a division algorithm based on the attributes of items, i.e., popularity and novelty, to divide the items into different sets. (2) We propose a feature extraction algorithm based on MI to characterize the recommendation attacks. (3) We conduct experiments on MovieLens dataset to verify the effectiveness of the proposed approach.

The rest of the paper is organized as follows. Section 2 describes the related definitions and theory used in this paper. Section 3 shows the details of the proposed feature extraction approach. Section 4 presents the experimental results and evaluations. The conclusions and future work are discussed in Section 5.

2. Related Definitions and Theory.

2.1. Related definitions.

The related definitions used in this paper are shown as follows.

Definition 2.1. *Popularity of Items (PoI). The popularity of item i, PoI_i, is defined as the ratio between the number of ratings given to item i by genuine users and the total number of genuine users in the training set R_{train}. It can be calculated as*

$$PopI_i = \frac{\sum\limits_{u \in R_g} \Gamma(r_{u,i})}{|R_g|}, \tag{1}$$

where, $R_g \subset R_{train}$ denotes the set of genuine users in the training set R_{train} and

$$\Gamma(r_{u,i}) = \begin{cases} 1, & r_{u,i} \neq \perp, \\ 0, & r_{u,i} = \perp, \end{cases} \tag{2}$$

where, $r_{u,i} \neq \perp$ denotes that item i is rated by user u and $r_{u,i} = \perp$ denotes that item i is not rated by user u.

Definition 2.2. *Novelty of Items (NoI). The novelty of an item refers to the degree to which it is unusual with respect to the user's normal tastes* [10]*. The novelty of item i, NoI_i, is defined as follows:*

$$NoI_i = \frac{1}{|R_g|} \sum_{u \in R_g,\, r_{u,i} \neq \perp} NoI_{u,i}, \tag{3}$$

where, $NoI_{u,i}$ denotes the novelty of item i to user u. $NoI_{u,i}$ can be calculated by the distance-based item novelty model in [11]*. $NoI_{u,i}$ is defined as follows:*

$$NoI_{u,i} = \frac{1}{N_j} \sum_{u \in R_g,\, r_{u,j} \neq \perp} (1 - sim(i,j)), \tag{4}$$

where, $N_j = \sum\limits_{j \neq i, r_{u,j} \neq \perp} \Gamma(r_{u,j})$ denotes the number of items rated by user u apart from item i and $sim(i,j)$ denotes the similarity between item i and item j. $sim(i,j)$ can be calculated as follows:

$$sim(i,j) = \frac{\sum_{u \in R_g} r_{u,i} \times r_{u,j}}{\sqrt{\sum_{u \in R_g} r_{i,k}^2} \sqrt{\sum_{k \in u} r_{j,k}^2}}. \tag{5}$$

2.2. Mutual information.

In the field of text mining, MI [12] is frequently used to measure the correlation between a term t and a category c. More details of MI can be found in [12].

3. Proposed Feature Extraction Approach.

Figure 1 shows the framework of the proposed feature extraction approach. As shown in Figure 1, the set of total items in the recommender system is first divided into four sets by the proposed division algorithm based on attributes of items. Then, features of users both in the training set and test set are extracted by the proposed feature extraction algorithm based on mutual information. In this process, user profiles in the original rating space are mapped into the feature space. The details of the proposed feature extraction approach will be discussed in the following subsections.

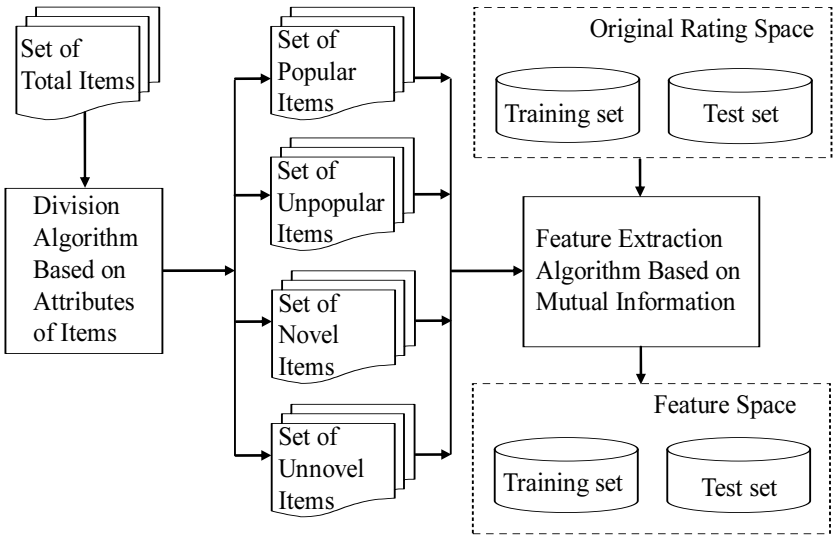

FIGURE 1. Framework of the proposed feature extraction approach

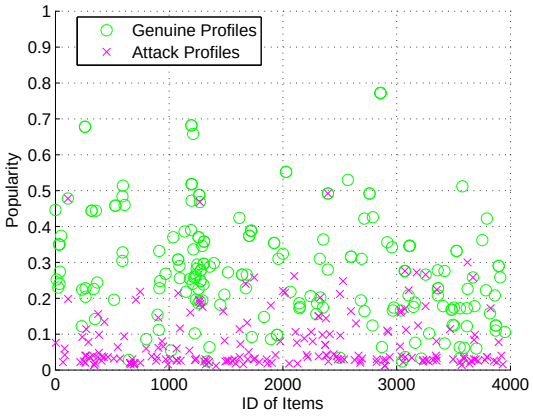

FIGURE 2. Rating distribution based on the popularity of items

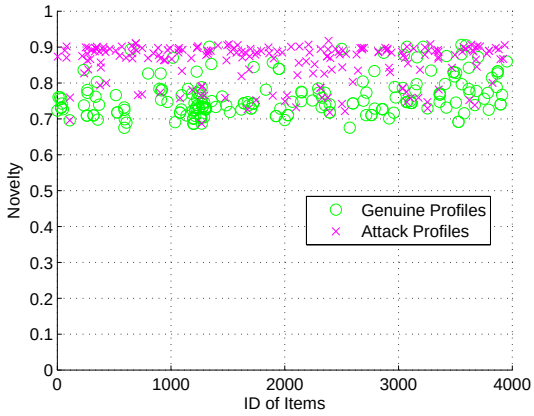

FIGURE 3. Rating distribution based on the novelty of items

3.1. **Division algorithm.** Genuine users rated items are usually according to their tastes and preferences. However, malicious users rated items might consider more about the purpose of the attacks. Therefore, different rating distribution based on the attributes of items, i.e., popularity and novelty, might be generated by the genuine profiles and attack profiles.

To show the difference between genuine and attack profiles in rating distribution, we show examples of popularity and novelty-based rating distribution in Figure 2 and Figure 3, respectively. The rating distribution is constructed by items rated in five genuine profiles and five attack profiles. The genuine profiles are randomly selected from the MovieLens dataset (discussed in Section 4.1). The attack profiles are constructed by the random attack model.

As shown in Figure 2 and Figure 3, the rating distribution of genuine profiles is different from that of attack profiles. More special, most rated items in genuine profiles are more popular than these in attack profiles. Most rated items in attack profiles are more novel than these in genuine profiles.

To capture these differences using the theory of MI, we first need to divide the items into different categories. That is, according to the popularity of items, we divide the items in the recommender system into popular set I_{pop} and unpopular set I_{unpop}. According to

the novelty of items, we divide the items in the recommender system into novel set I_{nov} and unnovel set I_{unnov}. Obviously, $I_{pop} \cup I_{unpop} = I_{nov} \cup I_{unnov} = I$, where I denotes the set of the total items in the recommender system.

To implement above process of division, we propose a simple algorithm based on the attributes of items as follows.

Algorithm 1 Division algorithm based on attributes of items

Input: I, k_{pop}, k_{nov}

Output: $\{ I_{pop}, I_{unpop}, I_{nov}, I_{unnov} \}$

1: Calculate PoI_i and NoI_i for each item $i \in I$

2: Sort the items according to their PoI in descending order

3: Let the top k_{pop} items belong to set I_{pop}

4: Let $I_{unpop} = I - I_{pop}$

5: Sort the items according to their NoI in descending order

6: Let the top k_{nov} items belong to set I_{nov}

7: Let $I_{unnov} = I - I_{nov}$

8: Return $\{ I_{pop}, I_{unpop}, I_{nov}, I_{unnov} \}$

As shown in Algorithm 1, parameters k_{pop} and k_{nov} determine the category of attributes, e.g., popular or unpopular for items in the recommender system where $1 \leq k_{pop} \leq |I|$ and $1 \leq k_{nov} \leq |I|$. For different recommender systems, k_{pop} and k_{nov} might be different. Therefore, in this paper we use five-fold cross-validation method to set them combining with the training set.

3.2. **Feature extraction algorithm.** After above division of items, we need to convert the rating database to a set of documents. In this process, users are regarded as terms, items are regarded as documents, and ratings rated by users are regarded as terms occurring in documents.

Let the category of the converted items determine the category of the corresponding documents. Therefore, we can use MI to measure the correlation between terms and categories of documents to capture the features of rating distribution for the detection of recommendation attacks. Based on this analysis, we propose 4 features as follows.

(1) *Mutual Information with Popular Items* ($MIPI$)

Feature $MIPI$ of user u characterizes the correlation between the items rated by user u and the items in set I_{pop}. It can be calculated as

$$MIPI_u = \log_2 \frac{A(u,I_{pop}) \times N}{(A(u,I_{pop}) + B(u,I_{pop})) \times (A(u,I_{pop}) + C(u,I_{pop}))}, \tag{6}$$

where, $A(u, I_{pop}) = |\{i | r_{u,i} \neq \perp, i \in I_{pop}\}|$ denotes the number of items rated by user u belonging to set I_{pop}, $B(u, I_{pop}) = |\{i | r_{u,i} \neq \perp, i \notin I_{pop}\}|$ denotes the number of items rated by user u but not belonging to set I_{pop}, $C(u, I_{pop}) = |\{i | r_{u,i} = \perp, i \in I_{pop}\}|$ denotes the number of items not rated by user u but belonging to set I_{pop}, and N=$|I|$ denotes the total number of items in the recommender system.

(2) *Mutual Information with UnPopular Items* ($MIUPI$)

Feature $MIUPI$ of user u characterizes the correlation between the items rated by user u and the items in set I_{unpop}. It can be calculated as

$$MIUPI_u = \log_2 \frac{A(u,I_{unpop}) \times N}{(A(u,I_{unpop}) + B(u,I_{unpop})) \times (A(u,I_{unpop}) + C(u,I_{unpop}))}, \tag{7}$$

where, $A(u, I_{unpop}) = |\{i | r_{u,i} \neq \perp, i \in I_{unpop}\}|$ denotes the number of items rated by user u belonging to set I_{unpop}, $B(u, I_{unpop}) = |\{i | r_{u,i} \neq \perp, i \notin I_{unpop}\}|$ denotes the number of items rated by user u but not belonging to set I_{unpop}, and $C(u, I_{unpop}) = |\{i | r_{u,i} = \perp, i \in I_{unpop}\}|$ denotes the number of items not rated by user u but belonging to set I_{unpop}.

(3) *Mutual Information with Novel Items* ($MINI$)

Feature $MINI$ of user u characterizes the correlation between the items rated by user

u and the items in set I_{nov}. It can be calculated as

$$MINI_u = \log_2 \frac{A(u,I_{nov})\times N}{(A(u,I_{nov})+B(u,I_{nov}))\times(A(u,I_{nov})+C(u,I_{nov}))}, \quad (8)$$

where, $A(u, I_{nov}) = |\{i|r_{u,i} \neq \perp, i \in I_{nov}\}|$ denotes the number of items rated by user u belonging to set I_{nov}, $B(u, I_{nov}) = |\{i|r_{u,i} \neq \perp, i \notin I_{nov}\}|$ denotes the number of items rated by user u but not belonging to set I_{nov}, and $C(u, I_{nov}) = |\{i|r_{u,i} = \perp, i \in I_{nov}\}|$ denotes the number of items not rated by user u but belonging to set I_{nov}.

(4) *Mutual Information with UnNovel Items* (*MIUNI*)

Feature *MIUNI* of user u characterizes the correlation between the items rated by user u and the items in set I_{unnov}. It can be calculated as

$$MIUNI_u = \log_2 \frac{A(u,I_{unnov})\times N}{(A(u,I_{unnov})+B(u,I_{unnov}))\times(A(u,I_{unnov})+C(u,I_{unnov}))}, \quad (9)$$

where, $A(u, I_{unnov}) = |\{i|r_{u,i} \neq \perp, i \in I_{unnov}\}|$ denotes the number of items rated by user u belonging to set I_{unnov}, $B(u, I_{unnov}) = |\{i|r_{u,i} \neq \perp, i \notin I_{unnov}\}|$ denotes the number of items rated by user u but not belonging to set I_{unnov}, and $C(u, I_{unnov}) = |\{i|r_{u,i} = \perp, i \in I_{unnov}\}|$ denotes the number of items not rated by user u but belonging to set I_{unnov}.

To implement the process of feature extraction, we propose a feature extraction algorithm based on the theory of MI as follows.

Algorithm 2 feature extraction algorithm based on mutual information

Input: $\{I_{pop}, I_{unpop}, I_{nov}, I_{unnov}\}$, u

Output: ($MIPI_u$, $MIUPI_u$, $MINI_u$, $MIUNI_u$)

1: Calculate $MIPI_u$ using Formula (6), set I_{pop}, and items are rated by user u

2: Calculate $MIUPI_u$ using Formula (7), set I_{unpop}, and items are not rated by user u

3: Calculate $MINI_u$ using Formula (8), set I_{nov}, and items are rated by user u

4: Calculate $MIUNI_u$ using Formula (9), set I_{unnov}, and items are not rated by user u

5: Return ($MIPI_u$, $MIUPI_u$, $MINI_u$, $MIUNI_u$)

As shown in Algorithm 2, u denotes one user belonging to training set or test set. ($MIPI_u$, $MIUPI_u$, $MINI_u$, $MIUNI_u$) denotes the feature vector constructed by the proposed features.

4. Experiments and Evaluations.

4.1. Experimental data and settings.

We use MovieLens dataset [13] to conduct the experiments. This dataset consists of 1,000,209 ratings on 3,952 movies by 6,040 users. All ratings are integer values between 1 and 5, where 1 is the lowest (disliked) and 5 is the highest (most liked). Each user in this dataset has rated at least 20 movies.

To create the training set, we randomly select 500 genuine profiles from the MovieLens dataset as samples of genuine profiles. Samples of attack profiles are generated by random attack, average attack, bandwagon attack, 20% AoP attack, 30% AoP attack, and 40% AoP attack with filler sizes $\{1\%, 3\%, 5\%, 10\%, 15\%\}$, respectively. Filler size [6] is defined as the ratio between the number of items rated by user u and the number of total items in the recommender system. We construct 10 attack profiles for each of the filler sizes corresponding to each of the above attack models. Therefore, the number of attack profiles for each attack model is 50.

Seven test sets are created. For each test set, we randomly select 700 genuine profiles from the remaining MovieLens dataset as samples of genuine profiles. Samples of attack profiles are generated by random attack, average attack, bandwagon attack, 20% AoP attack, 30% AoP attack, and 40% AoP attack with filler sizes $\{1\%, 3\%, 5\%, 10\%, 15\%\}$, respectively. In the first 6 test sets, we construct 20 attack profiles for each of the filler sizes corresponding to each of the above attack models. Therefore, the number of attack profiles for each attack model is 100 in each of the first 6 test sets. The 7th test set consists of mixture attack profiles. 5 attack profiles for each of the filler sizes corresponding to each of the above attack models are constructed. Therefore, the number of attack profiles

for each attack model is 25 in the 7th test set. This process is repeated 10 times and the average values of detection results are reported for the experiments.

Good performance of SVM-based classifier has been shown in [6]. Thus, to evaluate the performance of the proposed features we employ SVM to generate a classifier based on the proposed features (called SVM-Proposed) for the detection of recommendation attacks. Clearly, good detection performance of the generated classifier means high features' capability of characterizing recommendation attacks.

We compare the performance of SVM-Proposed with the following methods: (1) SVM-Williams [6]: An SVM-based supervised detection algorithm proposed in [6]. This is a representative supervised detection algorithm. Williams' features are used in this method. (2) PCA-VarSelect [4]: A representative unsupervised detection algorithm. This algorithm assumes that the number of injected attack profiles, k, is known in advance and returns the top-k profiles as attack profiles.

In our experiments, we use Libsvm 3.0 [14] to generate the classifier. The RBF is used as the kernel function. Five-fold cross-validation method with the training set is used to set the parameters of *gamma* and *cost* in Libsvm 3.0 and k_{pop} and k_{nov} in Algorithm 1.

To ensure the rationality of the results, the target item is randomly selected for each attack profile. The purposes of attacks (push or nuke) are also randomly determined.

4.2. **Evaluation metrics.** We use the standard binary classification measurements of sensitivity, specificity [6], and AUC (area under the ROC curve) [15] to evaluate the performance of the detection approaches used in our experiments.

4.3. **Experimental results and analysis.** Sensitivity of PCA-VarSelect, SVM-Williams, and SVM-Proposed on the 7 test sets is shown in Figure 4. As shown in Figure 4, PCA-VarSelect performs well when detecting the 1st, 2nd, 3rd, and 7th test sets. SVM-Williams performs well on the 1st, 5th, and 6th test sets. SVM-Proposed performs well when detecting all the test sets.

Note that, the only difference between SVM-Williams and SVM-Proposed is the features they used. However, the sensitivity of SVM-Proposed is higher than that of PCA-VarSelect and SVM-Williams on most of the test sets. These results illustrate that the proposed features can characterize the recommendation attacks, effectively.

Specificity of PCA-VarSelect, SVM-Williams, and SVM-Proposed on the 7 test sets is shown in Figure 5. As shown in Figure 5, the specificity of PCA-VarSelect and SVM-Proposed is a little higher than that of SVM-Williams. However, all the three methods

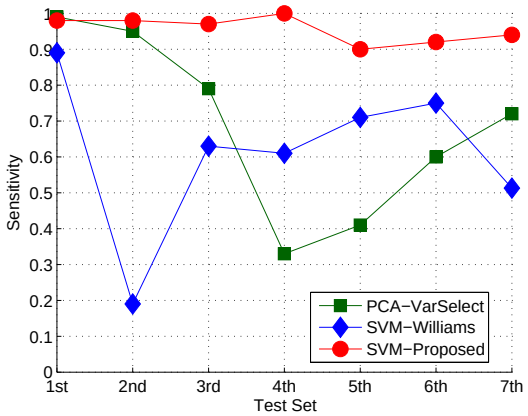

FIGURE 4. Sensitivity of PCA-VarSelect, SVM-Williams, and SVM-Proposed on the 7 test sets

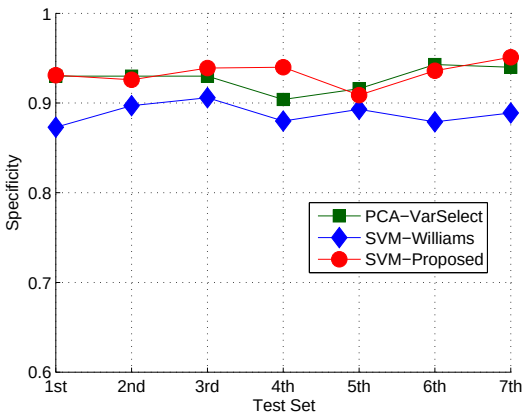

FIGURE 5. Specificity of PCA-VarSelect, SVM-Williams, and SVM-Proposed on the 7 test sets

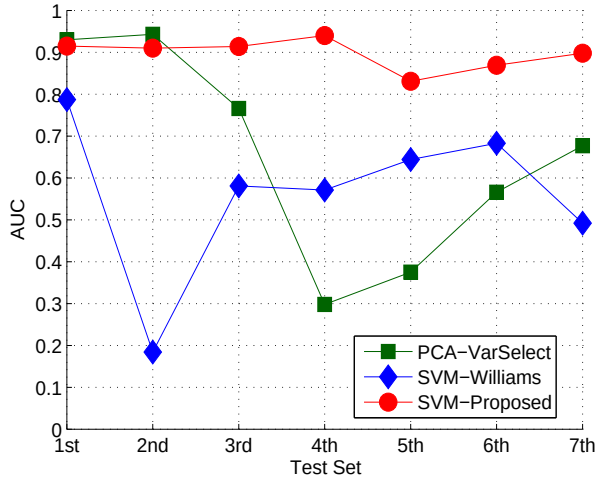

FIGURE 6. AUC of PCA-VarSelect, SVM-Williams, and SVM-Proposed on the 7 test sets

perform well on all the test sets. The reason for this phenomenon is that all the three methods can identify genuine profiles, effectively.

AUC of PCA-VarSelect, SVM-Williams, and SVM-Proposed on the 7 test sets is shown in Figure 6. As shown in Figure 6, SVM-Williams performs well only on a part of the test sets. This is due to the fact that Williams' features used in this method have limited capability of characterizing the recommendation attacks.

When detecting most of the test sets, SVM-Proposed outperforms PCA-VarSelect and SVM-Williams in terms of AUC. These results illustrate that SVM-Proposed can detect the recommendation attacks effectively when facing imbalanced test sets.

5. **Conclusions and Future Work.** In this paper, from the perspective of rating distribution a feature extraction approach for the detection of recommendation attacks based on MI is proposed. Based on the attributes of items, i.e., popularity and novelty, a division algorithm is proposed. This algorithm can divide the items in the recommender system into different sets. To characterize the recommendation attacks, a feature extraction algorithm is proposed based on the theory of MI. The experimental results on MovieLens dataset show that the proposed approach can effectively extract the features of recommendation attacks. In the future, we plan to extract more features of recommendation attacks from the perspective of rating relationship.

Acknowledgment. This work is supported by the Shandong Provincial Natural Science Foundation of China (No. ZR2014FP014).

REFERENCES

[1] J. M. Li, W. X. Wei, X. P. Hu and F. Sun, Multi-GPU based parallel collaborative filtering recommendation algorithm, *ICIC Express Letters*, vol.9, no.4, pp.1143-1151, 2015.

[2] S. K. Lam and J. Riedl, Shilling recommender systems for fun and profit, *Proc. of the 13th International Conference on World Wide Web*, New York, USA, pp.393-402, 2004.

[3] S. Zhang, Y. Ouyang, J. Ford and F. Makedon, Analysis of a low-dimensional linear model under recommendation attacks, *Proc. of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, Washington, pp.517-524, 2006.

[4] B. Mehta, H. Thomas and F. Peter, Lies and propaganda: Detecting spam users in collaborative filtering, *Proc. of the 12th International Conference on Intelligent User Interfaces*, Honolulu, Hawaii, pp.14-21, 2007.

[5] Z. Zhang and S. R. Kulkarni, Detection of shilling attacks in recommender systems via spectral clustering, *Proc. of the 17th International Conference on Information Fusion*, Salamanca, Spain, 2014.
[6] C. A. Williams, B. Mobasher and R. Burke, Defending recommender systems: Detection of profile injection attacks, *Service Oriented Computing and Applications*, vol.1, no.3, pp.157-170, 2007.
[7] R. Burke, B. Mobasher, C. Williams and R. Bhaumik, Classification features for attack detection in collaborative recommender systems, *Proc. of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Philadelphia, Pennsylvania, pp.542-547, 2006.
[8] M. A. Morid, M. Shajari and A. R. Hashemi, Defending recommender systems by influence analysis, *Information Retrieval*, vol.17, no.2, pp.137-152, 2014.
[9] F. Z. Zhang and Q. 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.
[10] N. Hurley and M. Zhang, Novelty and diversity in top-N recommendation – Analysis and evaluation, *ACM Trans. Internet Technology*, vol.10, no.4, pp.14:1-14:30, 2011.
[11] P. Castells, S. Vargas and J. Wang, Novelty and diversity metrics for recommender systems: Choice, discovery and relevance, *Proc. of International Workshop on Diversity in Document Retrieval at the 33rd European Conference on Information Retrieval*, Dublin, The Republic of Ireland, pp.29-36, 2011.
[12] Y. M. Yang and J. O. Pedersen, A comparative study on feature selection in text categorization, *Proc. of the 14th International Conference on Machine Learning*, Nashville, Tennessee, pp.412-420, 1997.
[13] *http://www.grouplens.org/node/12*.
[14] *http://www.csie.ntu.edu.tw/˜cjlin/libsvm/# download*.
[15] D. J. Hand and R. J. Till, A simple generalisation of the area under the ROC curve for multiple class classification problems, *Machine Learning*, vol.45, no.2, pp.171-186, 2001.