# Robust Recommendation Algorithm Based on User Suspicious Probability and Item Weight

Haihong Gao[1, a], Li Liu[1, b], Wenguang Zheng[1, c]

[a]haihonggaoo@163.com, [b]ll2012@tjut.edu.cn, [c]wenguangz@tjut.edu.cn

[1]School of Computer Science and Engineering, Tianjin University of Technology, Tianjin, China

**Abstract:** With the extensive development of recommendation technology, the threat of shilling attacks faced by the existing collaborative recommendation algorithms is also increasing sharply. To face more and more complex shilling attacks, this paper constructs a robust recommendation algorithm that can resist shilling attacks from the perspective of recommendation algorithm. Existing robust recommendation algorithms usually improve robustness by sacrificing some recommendation accuracy and reduce the recommendation accuracy. To solve this problem, this paper proposes a robust recommendation algorithm based on user suspicious probability and item weight. Firstly, we establish the relevance vector machine classifier according to user profile features to evaluate user suspicious probability in the database. Secondly, we construct singular value decomposition algorithm based on Hebbian learning and matrix factorization algorithm by integrating user suspicion information. Finally, a dynamic weighting scheme is used in combination with the above algorithm and the collaborative filtering algorithm based on item weight, and the above algorithms are mixed according to a certain weight to obtain a robust collaborative filtering algorithm SRICF. By adjusting the weight ratio, advantages of each algorithm are brought into play, thereby improving recommendation accuracy and robustness of the algorithm. Experimental results show that our algorithm has good prediction accuracy and robustness compared with other single algorithms.

**Keywords:** Shilling attacks, User suspicious probability, Relevance vector machine, Robust recommendation.

## 1 Introduction

Recommendation system has been widely used in various fields, to solve the problem of "information overload" [5]. However, due to its openness, anyone can influence the recommendation system through user behavior. Malicious attackers inject some false user behavior information into the recommendation system and try to change the recommendation results to seek benefits. This attack is called profile injection attack, or shilling attack [15]. Common attack types include random attack, average attack, bandwagon attack, etc. [6]. According to the purpose of attack, shilling attack can be divided into push attack and nuclear attack, which are used to increase or reduce the recommendations of target items, respectively.

ICIDC 2022, June 17-19, Qingdao, People's Republic of China

DOI 10.4108/eai.17-6-2022.2322755

A robust recommendation algorithm should be considered to deal with shilling attacks. However, in the face of more and more complex attacks, the existing methods usually improve the robustness by sacrificing some recommendation accuracy, which leads to relatively lower recommendation accuracy.

In this paper, we propose a robust recommendation algorithm named SRICF. The main contributions are summarized as follows:

(1) Establish a correlation vector machine classifier according to user profiles features, and evaluate the user suspicious probability in the database.

(2) Based on the user-item rating data, we construct a singular value decomposition algorithm based on Hebbian learning and matrix factorization algorithm.

(3) To reduce the impact of shilling attacks, we propose a new method to calculate the similarity between items.

## 2 Related works

The main purpose of robust collaborative filtering algorithm is to improve the robustness of the algorithm. Robustness means that it can predict well and has recommended stability even in the case of shilling attacks injection. O'Mahoney et al. proposed a new neighborhood selection and similarity weight transformation scheme, which compared the improved KNN with the traditional KNN algorithm and found that the effects of the two were similar in the absence of injection attacks, after adding the attack, the improved KNN is better than the traditional KNN algorithm in terms of accuracy and coverage, which proves that the improved algorithm has certain robustness [10]. Alonso et al. proposed a robust method based on matrix factorization to eliminate shilling attacks [2]. It detects abnormal changes in item predictions by obtaining the reliability value of the user's predictions for the item, avoiding the promotion of shilling predictions to the wrong recommendation, the results show that the method can effectively resist most of the existing attacks. Alostad et al. proposed a shilling attack defense detection algorithm SVM-GMM, which combines improved SVM and Gaussian mixture model (GMM) [1]. The experimental results show that this method has more advantages for the recall rate, accuracy rate and false positive rate among different attacks. Li et al. proposed an algorithm that first uses kernel principal component analysis to reduce the dimension of user rating matrix, and then uses fuzzy c-mean clustering method to distinguish normal users from shilling attack users [14]. Yi et al. proposed a new suspicious user measurement method, which uses the correlation vector machine classifier to identify and measure suspicious users, and uses the information of suspicious users to build a multi-dimensional trust model [13]. The experimental results show that the method has strong robustness. To verify the robustness of the algorithm in this paper, this paper is compared with several algorithms, and the experimental results show that our method can deal with the shilling attacks and has strong robustness.

## 3 Our approach

To ensure the quality of recommendations, we build a correlation vector machine classifier based on user profile features to evaluate the user suspicious probability. Then combined with the user suspicious information, it is used to construct singular value decomposition algorithm based on Hebbian learning and matrix factorization algorithm. Finally, a robust recommendation algorithm SRICF is designed by combining the singular value decomposition, the matrix factorization and the collaborative filtering.

### 3.1 The User Suspicious Probability Calculation

In this section, we introduce the formation process of the user suspicious probability. The process consists of two stages: the training process of the SVM-based classifier and the calculation process of the user suspicious probability.

In the training process, the training set includes real users and attack users marked with - 1 and 1 respectively. To generate training set to be computed, we use the WDMA [3] feature metrics.

During the calculation, we use the method in [7] to evaluate the user suspicious probability, as shown in Equation (1), where the values of parameters A and B are obtained using the SVM classifier.

$$P_i = \mathrm{P(y=1|x)} \approx \frac{1}{1+\exp\text{（}\mathrm{A}f+B\text{）}} \tag{1}$$

The training set is represented in the form of feature vectors. We can use the training set consisting of feature vectors to train correlation vector machines and generate SVM-based classifier. During computation, the set of user profiles to be computed is represented as a set of feature vectors. Therefore, we can use an SVM-based classifier to obtain the predicted probability of user classification (which we call it user suspicion degree).

### 3.2 SVD Algorithm Based on Hebbian Learning

Singular matrix decomposition (SVD) algorithm refers to decomposing a matrix into two orthogonal matrices and a diagonal matrix [12], and its matrix decomposition form is as follows.

$$D_{\mathrm{m}\times n} = U_{\mathrm{m\times k}} \Sigma_{k\times k} V_{n\times k}^{T} \tag{2}$$

Suppose D is an m×n-dimensional matrix, U and V are orthogonal matrices, and their dimension is m×k, n×k, ∑ is a diagonal matrix with dimension k×k. The biggest feature of SVD is that it transforms a high-dimensional data into a form of multiplying low-dimensional data, which realizes the dimensionality reduction of the data. However, if we directly use SVD in the recommendation system and use SVD to predict the rating matrix, the error will be large. Therefore, we have made improvements to the SVD algorithm. We used the Hebbian learning algorithm in SVD, and its main process is as follows.

Then, to predict the rating matrix $\hat{D}$ , the error e(x) is shown below.

$$\hat{D} = \hat{U}_i \times \hat{V}_j \tag{3}$$

$$e(x) = W_u \times (D - \hat{D}) \tag{4}$$

Where $W_u$ is the diagonal matrix calculated by Equation (5)

$$W_u = \mathrm{Clip}\left( \frac{1\text{-}TMF}{P_u} \right) \tag{5}$$

TMF is target model focus each user to the target item. Since the target item are generally rated high or low by the shilling attack user, the shilling attack user will pay a high degree of attention to the target item. $P_u$ is the suspicious probability of the user. Clip () truncates the value between 0 and 1. $W_u$ is the user's weight.

Since there is still an error in the prediction, we use the Hebbian learning algorithm to update the prediction rating matrix $\hat{U}_i$ , $\hat{V}_j$ as follows.

$$\hat{U}_i = \lambda \cdot \hat{V}_j \cdot e(x) \tag{6}$$

$$\hat{V}_j = \lambda \cdot \hat{U}_i \cdot e(x) \tag{7}$$

where λ is the learning rate. Although the prediction effect of the above formula is very good, it needs to go through many iterations. The possible reason is that the value of λ cannot be set too large, so some improvements are made in this paper.

The update function becomes:

$$\hat{U}_i = \hat{U}_i - \frac{\hat{U}_i - \lambda \cdot \hat{V}_j \cdot e(x)}{\sqrt{\Sigma_i^n (\hat{U}_i - \lambda \cdot \hat{V}_j \cdot e(x))}} \tag{8}$$

$$\hat{V}_j = \hat{V}_j - \frac{\hat{V}_j - \lambda \cdot \hat{U}_i \cdot e(x)}{\sqrt{\sum_j^n (\hat{V}_j - \lambda \cdot \hat{U}_i \cdot e(x))}} \tag{9}$$

### 3.3 Robust Recommendation Algorithm

Combining the singular matrix algorithm (SVD) based on Hebbian learning, the matrix factorization algorithm and the collaborative filtering algorithm based on item weight, a robust recommendation algorithm based on user suspicious probability and item weight is proposed. The formula of prediction level can be written as:

$$\hat{r}_{uv} = \alpha b_{uv} + \beta(p_u q_v^T) + \gamma(u_u v_v) \tag{10}$$

where $b_{uv}$ is the rating of the collaborative filtering algorithm based on item weight, $p_u q_v^T$ is the rating of the matrix factorization algorithm, $u_u v_v$ is the rating of the SVD algorithm based on Hebbian learning. The values of $b_{uv}$, $p_u q_v^T$, $u_u v_v$ are respectively related to the number of neighbors N of the item, the number of iterations R, and the hidden feature K. We use formula (11) (12) (13) to determine α, β, Υ.

$$\alpha = \frac{N}{N+R+K} \tag{11}$$

$$\beta = \frac{R}{N+R+K} \tag{12}$$

$$\Upsilon = \frac{K}{N+R+K} \tag{13}$$

In formula (14), $\overline{R}_v$ is the average value of item V, $\overline{R}_j$ is the average rating of item j, and sim (v, j) is the Pearson similarity as the weight factor of the item. At the same time, this paper proposes the calculation formula based on the weight similarity of popular items. There are k nearest neighbor items, and the first k items with high similarity are used as the nearest neighbor set of the item, and the accuracy of the recommendation result can be improved by adjusting the value of k.

$$b_{uv} = \overline{R}_v + \frac{\sum_{j \in neighbor(v)} (R_{uj} - \overline{R}_j) sim(v, j)}{\sum_{j \in neighbor(v)} | sim(v, j) |} \tag{14}$$

In order to achieve their own goals, the attacking users generally choose some popular items as their own filling items, forging user profiles (attack profiles) [11], making themselves look more like "normal users". In order to reduce the influence of shilling attacks and solve the interference of popular items on item similarity, this paper proposes a new method to calculate the similarity between items.

$$sim(i,j) = \frac{\sum_{t \in knn_i \cap knn_j} \frac{1}{\log(1+|knn_t|)}}{\sqrt{|knn_i|+|knn_j|}} \tag{15}$$

where *sim (i, j)* is the similarity between item *i* and item *j*, $|knn_i|$, $|knn_j|$ is the set of neighbors of items *i* and *j*, and $\frac{1}{\log(1+|knn_t|)}$ is the weight of popular items.

In formula (16), the weight calculated by formula (5) is substituted into the matrix factorization model and the regular term is added. The model changes as shown in formula (16), where $W_u$ is the diagonal matrix calculated by Eq. (5), $\lambda_p$, $\lambda_q$ are the regularization parameters of P and Q. Then use the stochastic gradient descent algorithm to obtain the update function in Equations. (17) and (18).

$$\min \|W_u \times (R - PQ^T)\|^2 + \lambda_p \|P\|^2 + \lambda_q \|Q\|^2 \tag{16}$$

$$P_{uk} = P_{uk} + \alpha(2W_u^2 e_{uj} q_{kv} - 2\lambda_p p_{uk}) \tag{17}$$

$$q_{kv} = q_{kv} + \alpha(2W_u^2 e_{uj} p_{uk} - 2\lambda_q q_{kv}) \tag{18}$$

# 4 Experiment

## 4.1 Experiment Setup

The experimental data set is MovieLens100K, and the data set contains 100,000 ratings of 1,682 items by 943 users, with ratings ranging from 1 to 5 integers. In this experiment, we divided the data set into training set and test set according to the ratio of 9:1. In order to prevent data over-fitting, the division of each data set is random. Although the results have certain changes, they are not too big.

In order to verify the robustness of the SRICF algorithm in this paper, we assume that the original user is a real user. Under the condition of different attack scale and filling size, we inject average attack into the data set, and choose push attack by default. The robustness of the algorithm is measured using two metrics: Absolute Mean Error (MAE) and Prediction Bias (PS).

### 4.2 Evaluation Metrics

To verify the robustness of the algorithm, the two metrics used: Absolute Mean Error (MAE) and Prediction Bias (PS) are shown below.

$$PS = \frac{1}{N} \sum_{i=1}^{U} \sum_{j=1}^{I} | r_{ij} - p_{ij} | \tag{19}$$

$$MAE = \frac{1}{N} \sum_{i=1}^{U} \sum_{j=1}^{I} | p'_{ij} - p_{ij} | \tag{20}$$

Where, $r_{ij}$ is the rating of users in the test set, $p_{ij}$ is the rating predicted according to the training set, $p'_{ij}$ is the predicted rating before being attacked, and N is the total number of ratings involved in the calculation.

### 4.3 Experiment Result and Analysis

**Influence of Experiment Parameters:** The number of nearest neighbors N, the number of iterations R and the number of hidden features K are the key factors affecting the SRICF algorithm. This chapter shows the effects of the number of nearest neighbors N, the number of hidden features K and the number of iterations R on the prediction accuracy of the algorithm.

In Figure1, for three attacks (random attack, average attack and bandwagon attack), we tested the influence of experimental parameters (number of nearest neighbors N, the number of iterations R and the number of hidden features K) on the prediction accuracy of the algorithm.

As can be seen from Figure1, the number of nearest neighbors N of the item should not be too large or too small. If it is too small, it cannot include enough nearest neighbors, which will inevitably reduce the prediction accuracy. Excessive assembly will increase the burden of calculation, which is not conducive to the further improvement of accuracy, and even damage the accuracy. When the number of nearest neighbors of the item N > 40, the value of MAE has not been significantly improved, and it is not necessary to further increase the value of N. It can be seen from Figure1 that the hidden feature number K has a great impact on the accuracy of the algorithm. In the face of average attacks, the performance effect of the algorithm is the best, followed by bandwagon attacks and random attacks. In the face of average attacks and bandwagon attacks, the value of MAE tends to be stable when the number of hidden features k > 25. As can be seen from Figure 1, the value of MAE decreases monotonically with the number of iterations R, but in the face of average attack, the number of iterations R > 600 has a slight upward trend.

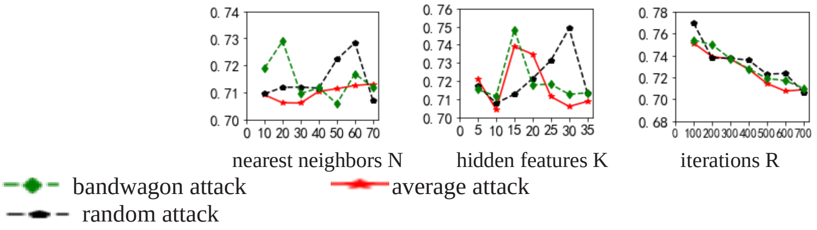

Figure 1: Influence of experiment parameters.

**Experiment Result:** First, this paper comprehensively evaluates the robustness of the SRICF algorithm on the MovieLens100K data set. The experiment adopts 3 × 4 × 4 design mode, attack model (random attack, average attack, bandwagon attack), attack size (1%, 3%, 5%, 10%) and filling size (3%, 5%, 7 %, 9%) correspond to a set of experimental configurations. The final data are taken from the mean of ten independent experiments, and Tables 1-3 are the experimental results.

It can be seen from the table that with the increase of attack intensity, the difference between MAE values in any table is no more than 0.03, which indicates that although the value of MAE has increased, the increase is not obvious, and the value of MAE has a downward trend in the case of average attack and bandwagon attack. In terms of the robustness of the algorithm, the PS value of the algorithm increases with the increase of the attack intensity, and this trend is particularly significant during bandwagon attacks. However, in actual situations, the attack intensity generally does not exceed 10%, otherwise it will lead to excessive attack costs, and its own attack behavior may also be exposed. As can be seen from the table, as the attack intensity increases, the SRICF algorithm can limit the PS value to a lower level, thereby weakening the degree of tampering of the attack target item sating by attackers. In addition, under the same parameter settings, the PS values of bandwagon attack is basically above random attack and average attack, indicating that bandwagon attack has the best attack effect, but is also accompanied by the highest attack cost.

Table1: The influence of average attack on the robustness of the algorithm.

| $p^{att}$ | $P^{fill}$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 3% | | 5% | | 7% | | 9% | |
| | MAE | PS | MAE | PS | MAE | PS | MAE | PS |
| 1% | 0.6907 | 0.2565 | 0.7002 | 0.2621 | 0.6972 | 0.2593 | 0.7236 | 0.2598 |
| 3% | 0.7114 | 0.2600 | 0.7062 | 0.2605 | 0.6962 | 0.2644 | 0.6941 | 0.2642 |
| 5% | 0.7030 | 0.2670 | 0.6924 | 0.2656 | 0.6859 | 0.2726 | 0.6793 | 0.2769 |
| 10% | 0.6790 | 0.2649 | 0.6713 | 0.2800 | 0.6505 | 0.2834 | 0.6572 | 0.2913 |

Table 2: The influence of random attack on the robustness of the algorithm.

| $p^{att}$ | $P^{fill}$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 3% | | 5% | | 7% | | 9% | |
| | MAE | PS | MAE | PS | MAE | PS | MAE | PS |
| 1% | 0.7204 | 0.2744 | 0.7098 | 0.2691 | 0.7271 | 0.2733 | 0.7260 | 0.2747 |
| 3% | 0.7202 | 0.2756 | 0.7181 | 0.2763 | 0.7207 | 0.2855 | 0.7244 | 0.3020 |
| 5% | 0.7167 | 0.2783 | 0.7343 | 0.2931 | 0.7294 | 0.3062 | 0.7360 | 0.3154 |
| 10% | 0.7291 | 0.2974 | 0.7524 | 0.3089 | 0.7543 | 0.3233 | 0.7658 | 0.3323 |

Table 3: The influence of bandwagon attack on the robustness of the algorithm.

| $p^{att}$ | $P^{fill}$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 3% | | 5% | | 7% | | 9% | |
| | MAE | PS | MAE | PS | MAE | PS | MAE | PS |
| 1% | 0.7000 | 0.2690 | 0.7329 | 0.2690 | 0.7325 | 0.2775 | 0.7052 | 0.2736 |
| 3% | 0.7244 | 0.2717 | 0.7058 | 0.2823 | 0.7194 | 0.2850 | 0.7227 | 0.2958 |
| 5% | 0.7045 | 0.2756 | 0.7117 | 0.2869 | 0.7198 | 0.2928 | 0.7239 | 0.3053 |
| 10% | 0.6878 | 0.3013 | 0.7042 | 0.3081 | 0.7127 | 0.3119 | 0.7168 | 0.3204 |

Next, we analyze the comparison of algorithm robustness in two cases: fixed $p^{fill}$= 3%, considered $p^{att}$ as a variable; fixed $p^{att}$ = 3%, considered $p^{fill}$ as a variable. R-MF [9], R-SIM [4], SIM [4], and R-TMF [8] were selected as baseline methods from existing methods. Among them, R-SIM is an algorithm that converts RDMA features into user trust with a Gaussian function, and R-TMF is an algorithm that converts TMF features into user weights by using a weight formula, and then substitutes them into the recommendation model for recommendation. Figure 2 shows the experimental results.

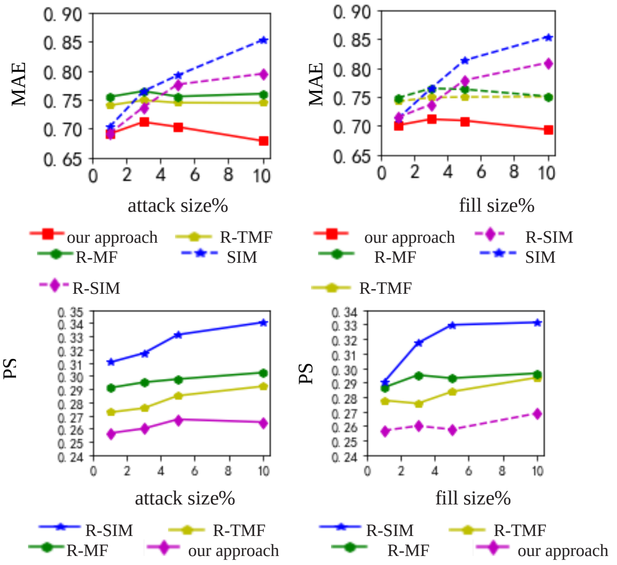

Figure 2: Comparison of algorithm robustness during average attack.

From Figure2, we can see that in terms of accuracy, our algorithm SRICF is the best, R-SIM is the worst, and R-TMF and R-MF are in between. This shows that our algorithm can meet the requirement of recommendation accuracy. In terms of robustness, our algorithm significantly outperforms other algorithms, indicating that our defense against shilling attack has a better effect. It can be seen from Figure 2 that during the attack size from 5% to 10%, our algorithm that prediction bias changes very little, which indicates that our algorithm has good stability, even under large-scale attacks, and it also maintains the ability to defend against shilling attack.

## 5 Conclusions

The robust recommendation algorithm (SRICF) based on user suspicious probability and item weight proposed in this paper combines the characteristics of matrix factorization algorithm (SVD, RMF) and item weight collaborative filtering algorithm. The feature combination constructs a weighting formula for fusion, and dynamically changes the weights of different algorithms by adjusting different parameters, so as to highlight the characteristics of each algorithm, so that the SRICF algorithm can meet different needs. Experimental results show that our algorithm improves the robustness of the algorithm without sacrificing recommendation accuracy. Future work will consider combining with other recommendation algorithms to improve the robustness while improving the recommendation accuracy.

## References

[1] Alostad, J.M. (2019). Improving the shilling attack detection in recommender systems using an svm gaussian mixture model. J. Journal of Information & Knowledge Management. (8):1950011.

[2] Alonso, S., Bobadilla, J. , Ortega, F. , & Moya, R. (2019).Robust model-based reliability approach to tackle shilling attacks in collaborative filtering recommender systems. *IEEE Access, 7(99),* 41782-41798.

[3] Burke, R. D. , Mobasher, B. , Williams, C. , et al. (2006). Classification features for attack detection in collaborative recommender systems[C] //Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 542-547.

[4] Fang, K.Q. & Wang, J. (2016). Defense Algorithm of shilling attack Based on matrix factorization model. J. Xiamen: Huaqiao University.

[5] Feng, S. , Meng, J. , & Zhang, J. (2021). News recommendation systems in the era of information overload. [J]. Web Eng,20(2):459- 470.

[6] Hao, Y. , & Zhang, F. (2020). An unsupervised detection method for shilling attacks based on deep learning and community detection. *Soft Computing* (8).

[7] Lin, H. T. , Lin, C. J. , & Weng, R. C. (2007). A note on platt's probabilistic outputs for support vector machines. *Machine Learning, 68(3),* 267-276.

[8] Liu, X. & Xiao, Y.Y. (2020). Robust recommendation algorithm based on limited target model focus. J. Tianjin University of technology, 36(01):1-5.

[9] Mehta, B., Hofmann, T., Nejdl, W. (2007). Robust collaborative filtering [C]//Proceedings of the 2007 ACM conference on Recommender systems. ACM, 49-56.

[10] O'mahony, M. P., Hurley, N. J., Silvestre, G. C. M. (2004). Utility-based neighbourhood formation for efficient and robust collaborative filtering.J. EC, 4: 260-261.

[11] Sundar, A. P., Li, F., Zou, X. , Gao, T. , & Russomanno, E. D. (2020). Understanding shilling attacks and their detection traits: a comprehensive survey. *IEEE Access, 8,* 171703-171715.

[12] Xun, Z., Jing, H., Huang, G., & Zhang, Y. (2015). Svd-based incremental approaches for recommender systems. J. Journal of Computer and System Sciences, 81(4):717-733.

[13] Yi, H. & Zhang, F. (2016). Robust recommendation method based on suspicious users' measurement and multidimensional trust. J. Journal of Intelligent Information systems,46(2): 349-367.

[14] Yi, H., Niu, Z., Zhang, F., et al. (2018). Robust recommendation algorithm based on kernel principal component analysis and fuzzy c-means clustering. J.Wuhan University Journal of Natural Sciences, 23(2):111-119.

[15] Zhang, F. & Zhou, Q., (2014). HHT-SVM: an online method for detecting profile injection attacks in collaborative recommender systems. *Knowledge Based System 65*(JUL.), 96-105.