Collaborative filtering-based recommendations against shilling attacks with particle swarm optimiser and entropy-based mean clustering

Anjani Kumar Verma

Department of Computer Science,
University of Delhi,
Delhi, India
Email: anjaniverma29@gmail.com

Veer Sain Dixit*

Department of Computer Science,
ARSD College,
University of Delhi,
Delhi, India
Email: veersaindixit@rediffmail.com
*Corresponding author

Abstract: Recommender system (RS) in the present web environment is required to gain the knowledge of the users and their commitments such as like and dislike about any items available on the e-commerce sites. Movie recommendations are one of such type in which shilling attack is increasing day by day, this will destroy or abruptly disturb the meaning of the data when recommended to others. Also, the hazards of shilling attacks degrade the performance of web recommendations. Hence, to address this issue the paper, collaborative filtering (CF)-based hybrid model is proposed for movie recommendations. The entropy-based mean (EBM) clustering technique is used to filter out the different clusters out of which the top-N profile recommendations have been taken and then applied with particle swarm optimisation (PSO) technique to get the more optimised recommendations. This research is focused is on getting secure recommendations from different recommender systems.

Keywords: collaborative filtering; entropy-based mean; EBM; recommender system; particle swarm optimiser; shilling attack.

Reference to this paper should be made as follows: Verma, A.K. and Dixit, V.S. (2023) 'Collaborative filtering-based recommendations against shilling attacks with particle swarm optimiser and entropy-based mean clustering', *Int. J. Information and Computer Security*, Vol. 20, Nos. 1/2, pp.133–144.

Biographical notes: Anjani Kumar Verma is currently designated as an Assistant Professor in Cluster Innovation Centre, University of Delhi. He has been working as an Assistant Professor in different College of Delhi University in the specialisation of Computer Science. He has a total of eight years of experience and has published papers in international conferences and journals. He is currently pursuing his PhD from the Department of Computer Science,

Copyright © 2023 Inderscience Enterprises Ltd.

University of Delhi. His area of working includes information security and expert systems. Earlier, he completed his BSc (Hons.) in Computer Science and MSc Computer Science from University of Delhi.

Veer Sain Dixit received his PhD in Computer Science. He is an Associate Professor in the Department of Computer Science, ARSD, University of Delhi. He has about 18 years of teaching and research experience and has published about 40 papers in national/international journals/conferences. His current research area is artificial intelligence like web recommender systems, web mining, machine leaning, etc. He has authored books on Computer Science subjects for national and international publishers like Narosa, etc. He has served as the Academic Council member DU, member governing body, member DRC, member PG Committee of courses, Examination Deputy superintendent, worked in various university committees and college committees as well as Assistant Coordinator IGNOU, coordinated Department of Computer Science, ARSD College for 14 years, admission in charge for 14 years. He is a life member of IETE.

1 Introduction

Recommender systems are the most important way to know the user, what they like or dislike, and what their opinions towards any items are. In the same manner, there are different varieties of users are on many e-commerce sites who are buying or purchasing the items. There are positive as well as negative users are on such sites. Negative user means, the malicious users and/or parties. We have focused on a collaborative filtering-based recommendation system, as it is highly vulnerable to profile injection. Nowadays, they make a great success in many e-commerce applications.

1.1 Shilling attack

Recommender systems based on collaborative filtering are vulnerable to 'shilling attacks' due to their open nature (Katarya and Verma, 2018).

Shilling attack or profile injection attacks that refer to the promotion of the attacker's item or demotion of his opponent's item. In such attacks, the fictitious user creates a large number of attack profiles using any automated tool and injects them to gain system benefits.

As a result, the system would generate recommendations that are irrelevant to the genuine user due to which he might lose his trust in that particular recommender system. Insertion of the fake profiles can be done either using some tool or manually. If an attacker wants to insert a large number of fake profiles in the system, then it is not an optimal way to insert them manually. In that case, he should use some automated tool to generate and insert them into the system. Profile injection attack can be categorised into two types, i.e., push attack or nuke attack (Katarya and Verma, 2018). The type of attack depends on the aim of the attacker. If he wants to promote his item then it is pushed attack and if he wants to demote the item of his opponent then the nuke attack is mounted.

1.2 Elements of attack

The bias in the recommender systems is prone to attacks injection that results from biased data that can be added to the system and spoil the predictions of a target item. These attacks can be categorised based on the following elements:

- *Knowledge required for mounting an attack:* Some efforts are required to mount an attack against the system. Gathering knowledge about the system is one such effort.
- *Attack size:* The size of the attack is the number of unscrupulous profiles inserted by the attacker. Generating and injecting fake profiles automatically require less cost and effort.
- *Profile size:* It is defined ratings under attack profile. Providing ratings to an item requires less cost as compared to that required for creating the attack profile.
- *The intent of an attack:* The attacker assigned a rating to the target items based on the purpose of the attack. A malicious user may inject biased profiles to make the less popular item to be more likely 'push' and the most popular item to be less likely 'nuke'.

1.3 Collaborative filtering

There are two challenges for collaborative filtering recommender systems (CFRs).

1 To improve the scalability of CF algorithms. Today, there is a demand for searching hundreds of millions of most similar neighbours. But, the existing CF algorithms have performance issues; they take more time to search for relevant information from such a large amount of available information.

2 To improve the recommendation quality. The accuracy of the system is important. It should recommend only those items that are relevant to the user to maintain the trust of the user on it.

- *Memory-based collaborative filtering:* Memory-based algorithms use the input user item matrix to predict the ratings for an item (Linden et al., 2003). These systems utilise various statistical methods.
- *Model-based collaborative filtering:* In this approach, the system uses training data and learns the complex patterns, and then makes predictions for the test users depending upon the learned models (Vucetic and Obradovic, 2005). The dataset is partitioned into training and testing then the models is trained using the training dataset and the performance of the models is analysed on the testing dataset.
- *Hybrid CF:* This approach combines the advantages of both memory and model-based CF algorithms (Su et al., 2007).

The main motivation for writing this paper has been drawn from different studies. In Bellogin et al. (2014), it is pointed out that if people's ratings are noisy, inconsistent, and biased then it degrades the quality of the product for future recommendations. In Li et al. (2013), too many noisy ratings are determined that could distort users' preference which results in neighbours that are unlike-minded will lose the quality of recommendations. In Katarya and Verma (2017), a system (K-means cuckoo) is proposed to perform the

experiments on the Movielens dataset and it was found that the accuracy has improved when compared with the existing methods.

There are many challenges associated with recommender systems like data sparsity, shilling attack, cold start problem (Rosli et al., 2015), black sheep-grey sheep problem, and scalability. Here, we are addressing the Shilling Attack problem in the proposed system.

The major highlights of this paper are as follows:

- we proposed a hybrid approach using entropy-based mean clustering technique and particle swarm optimiser
- the clustering process has been performed using the entropy-based mean clustering technique
- particle swarm optimisation is applied to get the optimised recommendations out of top-n recommendations
- Mean absolute error (MAE) gives better results compared to existing methods
- analysed the behaviour of the proposed model using Movielens 1M Dataset (https://grouplens.org/datasets/movielens/).

The paper is organised as follows: Section 2 briefly explains related works that have been carried out on collaborative recommender systems. In Section 3, the proposed method is explained for the movie recommender system. In Section 4, results and discussions are analysed. In the end, the conclusions with a summary and highlighted the future subsequent work are explained in Section 5.

2 Related works

After rigorous study about RS, it was noticed that most of the researchers are having a focus on the CF-based approach on different challenges.

Studies of various researchers' works are as follows: in Katarya and Verma (2018), a movie-based collaborative recommender system, grey wolf optimiser (GWO) algorithm, and fuzzy C-means (FCM) clustering technique are used with Movielens dataset (Katarya and Verma, 2016). This approach is performed well concerning accuracy and precision. It has been noticed that the hybrid approach FCM with GWO has 0.68 MAE which could not give better results than ours.

Mehta and Nejdl (2009) explained attack models that can be readily detected using statistical detection techniques. They used the principal component analysis (PCA)-based clustering strategy on the Movielens dataset by obtaining 90% precision in the detection of average and other attack models.

In Chakraborty and Karforma (2013), the outlier detection problem is discussed which is more close to profile injection attacks in terms of resemblance, one of the papers in which the author has used rating dataset with partition around medoids (PAM) clustering algorithm in detecting the injected profiles.

In Bilge et al. (2014), a binary decision tree (BDT) is proposed which has used the K-means clustering algorithm to locate the fake attack profiles.

A recent survey paper (Yera and Martinez, 2017) has used traditional fuzzy tools, which accurately process the information in RSs. However, there are several research gaps in trends in RSs.

In Castro et al. (2018), natural noise management for groups based on fuzzy tools (NNMG-FT) has used fuzzy profiling, global noise management, and local noise management; there are noise detection and noise correction fuzzy tools that allow classification of the ratings into noisy or not noisy.

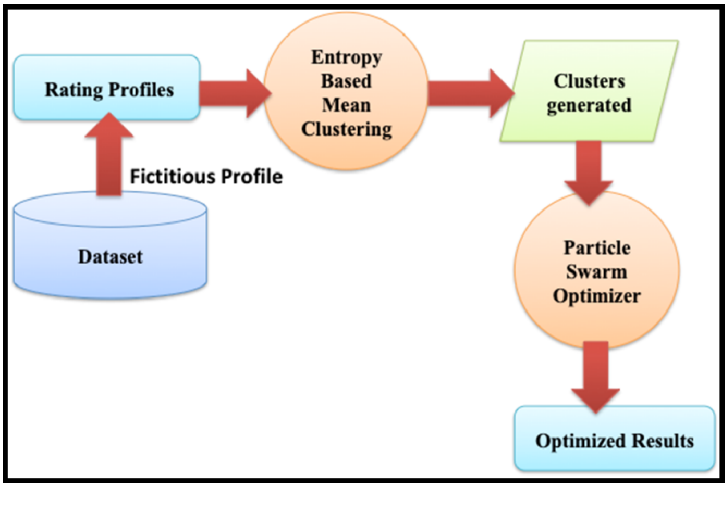
In Rosli et al. (2015), temporal collaborative filtering (TCF) approaches are used in the recommendation. Pearson correlation coefficient (PCC) as a measurement is used to obtain the similarity between the groups of the cluster. Then, the top-N highest predictive items are selected to recommend the target user.

3 Proposed method

In this section, the detailed framework of the proposed method is introduced in two stages that include the hybrid approach, i.e., entropy-based mean clustering and particle swarm optimisation algorithms and applied to the movie review dataset.

We have chosen this method because these will efficiently filtered profiles from maliciously injected profiles and EBM is more effective than K-means clustering because it handles empty clusters. PSO is used for better-optimised results. Overall, we can say that our proposed method is resisting attacks over other methods that are compared with more effectively as we have used the average attack model with that. In the first phase, pre-processed rating profiles have been obtained and applied on the EBM method, from there top-N recommended rating profiles are obtained. After that these top-N profiles are applied on the PSO method and optimised profiles for recommendation are produced in the second phase, as shown in Figure 1.

Figure 1 The framework of the proposed method (see online version for colours)



This paper has used to describe the attack and non-attack profile by the algorithm.

Algorithm: Filtering rating profiles

Initialise dataset: Movielens 1M dataset

1 ***do***

2 ***for*** *all ratings*

3 *if the rating is nil or less than equal to 2*
then the profile is not significant call as ***"attack profile"***

4 *else*
Store the profile in the database for the EBM process

5 ***end***

6 ***end***

//Entropy-based mean clustering process

7 ***do***

8 ***for*** *all ratings after the EBM process*

9 *if the rating is top-n*
then store the rating for the PSO process

10 *else*
remove the rating profile

11 ***end***

12 ***end***

//Particle swarm optimisation process

13 ***do***

14 ***for*** *all top-n ratings*

15 *if the ratings are best optimal after the PSO process*
then store it as ***"optimised personalised rating"*** *for recommendations*

16 *else*
repeat the process till the optimised will not be achieved.

17 ***end***

18 ***end***

3.1 Entropy-based mean clustering

The proposed EBM algorithm works in the following stages:

- In stage 1, minimum points are computed for each seed in the dataset and arranged in the order of their seed entropy (Ramakrishnaiah et al., 2012).
- In stage 2, the candidate set is made which is distinct without duplicate elements.
- In stage 3, the clustering was applied using Euclidean distance whereas the remaining resided elements were placed in the native elements.

3.2 Particle swarm optimisation

In particle swarm optimisation, the main idea is to obtain the global position (optimal) in d-dimensional space having optimal fitness value. While searching initially there is a need to initialise, the position of each particle in a random fashion. Every time the velocity is computed as per PSO; the position will be updated with its velocity. Once the predefined criterion is reached, iterations stop. There are two kinds of positions (best personal position and global position respectively).

The movement to the best personal position and the best global position while movement from current velocity is done. Once we get the velocity, the next position of this particle could be obtained.

In PSO, for each particle, the representation of position is defined as: position-vector, $X_i = (x_{i1}, x_{i2}, \ldots, x_{ij}, \ldots)$ where i is the particle index, j is the dimension index; the velocity is represented as, $V_i = (v_{i1}, v_{i2}, \ldots, v_{ij}, \ldots)$; the best personal position is represented by position-vector $P_i = (p_{i1}, p_{i2}, \ldots, p_{ij}, \ldots)$; and the global best position is represented by position-vector $G = (g_1, g_2, \ldots, g_j, \ldots)$. During iteration t, the new velocity in equation (1) and the new position in equation (2) are updated as follows:

$$\mathrm{Vi}(t+1) = \omega \cdot \mathrm{Vi}(t) + c1 \cdot \mathrm{rand} \cdot (\mathrm{Pi} - \mathrm{Xi}(t)) + c2 \cdot \mathrm{rand} \cdot (\mathrm{G} - \mathrm{Xi}(t)) \tag{1}$$

$$\mathrm{Xi}(t+1) = \mathrm{Xi}(t) + \mathrm{Vi}(t+1) \tag{2}$$

In the equation above, ω is denoted as the inertia factor.

Inertia factor in the formula controls the impact of current velocity to the next velocity. 'rand' in the given equation is a random number distributed in [0,1]; to keep the randomness and diversity of the population.

c1 and c2 are weighting coefficients that pull the particle towards the personal best position or the global best position (Abraham et al., 2006). Now how to obtain the next iteration's position of each particle could be explained through the equation above.

The global best position gets improved again and again in the PSO method. PSO will end its execution until the maximum number of iterations reached or it stops the iterations after several rounds of no improvement.

In BPSO for every dimension, there are only two possible values – 0 (i.e., bad in our case) and 1 (i.e., good in our case). A sigmoid function is taken to transform Vi to the range of (0, 1) in equation (3) (Xue et al., 2012). The position update of BPSO is defined as per the in equation (4) below:

$$SW(V_i(t+1)) = \frac{1}{1 + e^{-V_i(t+1)}} \tag{3}$$

$$X_i(t+1) = \begin{cases} 1, & \text{if } rand < S(V_i(t+1)) \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where rand denotes random number [0, 1].

4 Results and discussion

In this section, experiments are performed with the proposed model. All these experiments are performed on the system, which has a configuration of 8 GB RAM and a WEKA 3.8 (https://www.cs.waikato.ac.nz/) environment.

4.1 Dataset

We have taken a Movielens 1M Dataset (https://grouplens.org/datasets/movielens/) with a rating scale of 1 to 5. The experimental dataset contains 10,000 ratings with 200 users on 50 items for our experiment.

The average attack is used and designed the data in the range of 5% to 25% filler size to analyse the results with the dataset. As compared to other attack models in shilling attack, average attack model is the efficiently detect the attacks over user-item matrix. The result gets improved in terms of error rate on different filler items.

4.2 Measures

To Measure the accuracy we have used the following metrics.

Mean absolute error

MAE is defined as in equation (5) below:

$$MAE = \frac{\sum |\, p - r \,|}{M} \tag{5}$$

where M is the total predicted movies, p represents the predicted rating and r is the actual rating. Lower values are better for results.

Standard deviation

Standard deviation (SD) can be calculated as in equation (6) below:

$$SD = \frac{\sum_{i=1}^{k} \left\{ \sum_{j=1}^{no.\ of\ elements\ in\ i} \left(\sum_{l=1}^{m} \sqrt{\frac{(expect\ l - mean\ Kn\ l)^2}{m}} \right) \right\}}{no.\ of\ elements\ in\ i} \tag{6}$$

Precision

The precision has been calculated as in equation (7) below:

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

4.3 Results

The comparisons that were performed on the already existing system has shown that, while increasing the clusters, the MAE with the proposed method gives a lesser value than the existing method as can be seen on the large size clusters in Table 1.

Table 1 Comparison of MAE with number of cluster on different methods

# cluster (k)	*PCA + SOM (Katarya and Verma, 2018)*	*K-means + cuckoo (Katarya and Verma, 2017)*	*GWO + FCM (Katarya and Verma, 2018)*	*PCA + K-means (Katarya and Verma, 2017)*	*EBM + PSO*
	Existing	*Existing*	*Existing*	*Existing*	*Proposed*
64	1.96	0.6842	0.68	0.64	0.6222

Then, SD is computed with a proposed method on increasing the number of clusters, it is also obtained as a lesser value than the existing once on large size clusters see in Table 2.

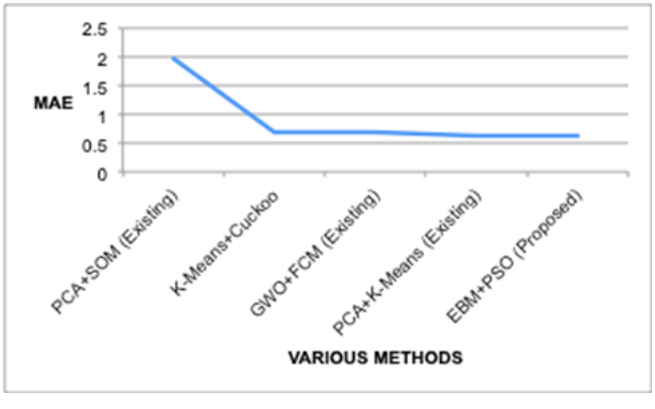
Table 2 Comparison of SD with number of cluster on different methods

# cluster (k)	*PCA + SOM (Katarya and Verma, 2018)*	*K-means + cuckoo (Katarya and Verma, 2017)*	*GWO + FCM (Katarya and Verma, 2018)*	*PCA + K-means (Katarya and Verma, 2017)*	*EBM + PSO*
	Existing	*Existing*	*Existing*	*Existing*	*Proposed*
64	1.15	0.1094	0.54	0.73	0.0923

Finally, it can be seen that the results have shown that the proposed model is outperformed over existing once.

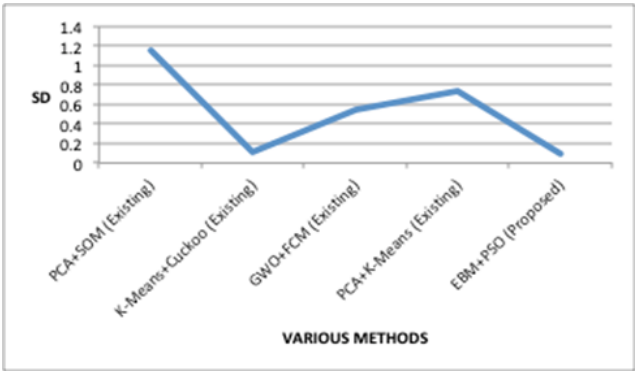
With the initial smaller number of clusters, our model is not so good but as the number of clusters increases in sufficient number, our system outperformed the existing one. The graph has been plotted between various methods on MAE that shows that MAE gives better results on the proposed method that is shown in Figure 2.

Figure 2 The plot between various methods for MAE (see online version for colours)



Now, as the number of clusters increases in sufficient number, our system outperformed the existing one. The graph has been plotted between various methods on SD that shows that SD gives better results on the proposed method that is shown in Figure 3.

Figure 3 The plot between various methods for SD (see online version for colours)



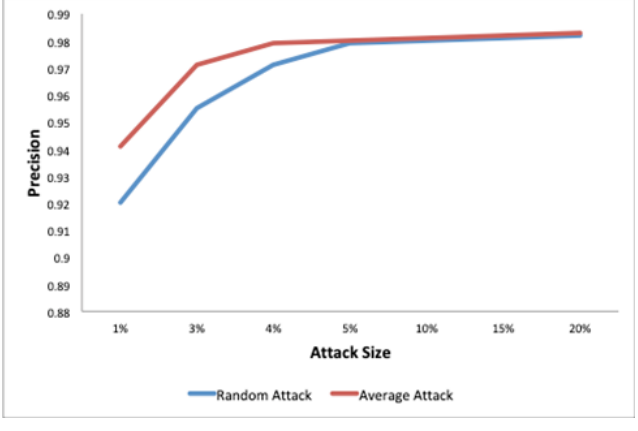
The results are also analysed by taking different attack models such as random and average attacks with different filler sizes taken as 0% to 20% (see Table 3). It has been noticed that the average attack model has smooth precision compare to other shilling attack models.

Table 3 Precision of random and average attack model over different attack size

Attack size	*Precision random attack*	*Precision average attack*
1%	0.92	0.941
3%	0.955	0.971
4%	0.971	0.979
5%	0.979	0.98
10%	0.98	0.981
15%	0.981	0.982
20%	0.982	0.983

The graph has been depicted in Figure 4 shows the overall precision of the average attack that is more efficiently detect the shilling attack over random attack type.

Figure 4 The plot between average, random attack vs. precision (see online version for colours)



5 Conclusions

In this paper, a hybrid of EBM and PSO is applied to the dataset to achieve improved movie recommendations. The performance has been measured through MAE and SD. The experiments indicate that the discussed approach provides high performance in the case of a large number of clusters and was capable of providing recommendations. The metrics for the evaluation (for a given number of clusters) come out to be lesser than those of other existing methods. Finally, the precision has been evaluated between average and random attack and we get the better result in the case of average attack. As far as the future work is a concern, we will apply security check on a different available recommender system such as Amazon and Netflix to predict the behaviour under attacked and non-attacked environment.

Acknowledgements

A special thanks to Dr. Veer Sain Dixit as my supervisor for his constructive suggestions about entropy-based clustering for the optimisation and the authors have used his remark on preparing this work.

References

Abraham, A., Guo, H. and Liu, H. (2006) 'Swarm intelligence: foundations, perspectives and applications', in Nedjah, N. and Mourelle, L.M. (Eds.): *Swarm Intelligent Systems. Studies in Computational Intelligence*, Vol. 26, Springer, Berlin, Heidelberg, https://doi.org/10.1007/978-3-540-33869-7_1.

Bellogin, A., Said, A. and de Vries, A.P. (2014) 'The magic barrier of recommender system – no magic, just ratings', in *International Conference on USR Modeling, Adaptation, and Personalization*, Springer, pp.25–36.

Bilge, A., Ozdemir, Z. and Polat, H. (2014) 'A novel shilling attack detection method', *Procedia Computer Science*, Vol. 31, pp.165–174.

Castro, J., Yera, R. and Martinez, L. (2018) 'A fuzzy approach for natural noise management in group recommender systems', *Expert Systems with Applications*, Vol. 94, pp.237–249.

Chakraborty, P. and Karforma, S. (2013) Detection of profile-injection attacks in recommender systems using outlier analysis', *Procedia Technology*, Vol. 10, pp.963–969.

Katarya, R. and Verma, O.P. (2016) 'A collaborative recommender system enhanced with particle swarm optimization technique', *Multimedia Tools and Applications*, Vol. 75, No. 15, pp.9225–9239.

Katarya, R. and Verma, O.P. (2017) 'An effective collaborative movie recommender system with cuckoo search', *Egyptian Informatics Journal*, Vol. 18, No. 2, pp.105–112.

Katarya, R. and Verma, O.P. (2018) 'Recommender system with grey wolf optimizer and FCM', *Neural Computing and Applications*, Vol. 30, No. 5, pp.1679–1687.

Li, B., Chen, L. and Zhang, C. (2013) 'Noisy but non-malicious user detection in social recommender systems', *World Wide Web*, Vol. 16, Nos. 5–6, pp.677–699.

Linden, G., Smith, B. and York, J. (2003) 'Amazon.com recommendations: item-to-item collaborative filtering', *Internet Computing*, Vol. 7, No. 1, pp.76–80, IEEE.

Mehta, B. and Nejdl, W. (2009) 'Unsupervised strategies for shilling detection and robust collaborative filtering', *User Modeling and User-Adapted Interaction*, Vol. 19, Nos. 1–2, pp.65–97.

Movielens 1M Dataset [online] https://grouplens.org/datasets/movielens/ (accessed 21 January 2019).

Ramakrishnaiah, V.J., Rao, D.K.R.H. and Prasad, D.R.S. (2012) 'Entropy based mean clustering: a enhanced clustering approach', *The International Journal of Computer Science and Applications*, Vol. 1, No. 3, pp.1–9, ISSN-2278-1080.

Rosli, A.N., You, T., Ha, I., Chung, K.Y. and Jo, G.S. (2015) 'Alleviating the cold-start problem by incorporating movies Facebook pages', *Cluster Computing*, Vol. 18, No. 1, pp.187–197.

Su, X., Greiner, R., Khoshgoftaar, T. and Zhu, X. (2007) 'Hybrid collaborative filtering algorithms using a mixture of experts', in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, IEEE Computer Society, pp.645–649.

Vucetic, S. and Obradovic, Z. (2005) 'Collaborative filtering using a regression-based approach', *Knowledge and Information Systems*, Vol. 7, No. 1, pp.1–22.

Weka 3.8 [online] https://www.cs.waikato.ac.nz/ (accessed 21 June 2018).

Xue, B., Zhang, M. and Browne, W.N. (2012) 'Single feature ranking and binary particle swarm optimization based feature subset ranking for feature selection', in *Proceedings of the Thirty-Fifth Australasian Computer Science Conference*, Melbourne, Australia, Vol. 122, pp.27–36.

Yera, R. and Martinez, L. (2017) 'Fuzzy tools in recommender systems: a survey', *International Journal of Computational Intelligence Systems*, Vol. 10, No. 1, pp.776–803.