TEM Journal. Volume 13, Issue 2, pages 1352-1361, ISSN 2217-8309, DOI: 10.18421/TEM132-49, May 2024.

Collaborative Filtering Recommender System for Online Learning Resources with Integrated Dynamic Time Weighting and Trust Value Calculation

Pengyu Guo [1], Mohd Khalid Mohamad Nasir [1], Yishuai Xu [2]

[1] *Faculty of Education, National University of Malaysia, Bangi, 43600, Malaysia*
[2] *Department of Library and Information Science, University of Malaya, Kuala Lumpur, 50603, Malaysia*

***Abstract* – Traditional educational models struggle to meet the demands of students seeking personalized online learning resources (OLRs). Collaborative filtering (CF) algorithms are widely employed for personalized OLR recommendations, yet they encounter issues such as poor scalability, cold start, and sparse data issues. In response, an enhanced CF algorithm is proposed, incorporating a fusion of time weighting and a credibility selection strategy. Initially, interactions and ratings among learners are analyzed. Subsequently, the algorithm integrates learner similarity and trust, calculating the credibility value weight between learners. Dynamic time weighting is then introduced separately into CF algorithms based on OLRs and learners, respectively. Ultimately, the algorithm predicts learner ratings for unknown OLRs. Experimental comparisons demonstrate that the performance metrics of the hybrid algorithm presented in this paper show significant improvement over traditional and other improved algorithms. It exhibits enhanced rating prediction accuracy, facilitating precise recommendations of personalized OLRs to learners.**

***Keywords* – Online learning resources, collaborative filtering, personalized recommendation, dynamic time weighting, trustworthy selection strategy.**

DOI: 10.18421/TEM132-49
https://doi.org/10.18421/TEM132-49

Corresponding author: Mohd Khalid Mohamad Nasir,
Centre of STEM Enculturation, Faculty of Education, National University of Malaysia, Bangi, 43600, Malaysia
Email: mdkhalid@ukm.edu.my

Received: 16 January 2024.
Revised: 15 April 2024.
Accepted: 24 April 2024.
Published: 28 May 2024.

1. Introduction

The rapid evolution of the Internet has made searching and accessing information exceptionally convenient. However, as the volume of available data increases, managing information becomes challenging and may lead to issues like "information explosion" and "information overload" [1]. The latter refers to the incapacity of recipients or processors to handle complex and abundant internet information, hindering the precise and swift retrieval of personal information needs. Faced with explosive data growth, learners find it increasingly difficult to locate valuable information that caters to their individual requirements. To enhance information filtering and elevate user service quality, recommendation systems (RS) have emerged, swiftly capturing widespread attention in academic circles [2].

In recent years, Massive Open Online Courses (MOOCs) have gained popularity among students, with platforms such as EdX, Coursera, and Khan Academy being widely utilized [3]. Despite the convenience offered by online learning, challenges persist in the online learning resource (OLR) recommendation process. Firstly, the diverse array of resource types leads to fuzzy categorization due to multiple labels attached to the same resource. Secondly, online learning platforms fail to fully exploit potential relationships among learner characteristic information, making personalized recommendations a formidable challenge.

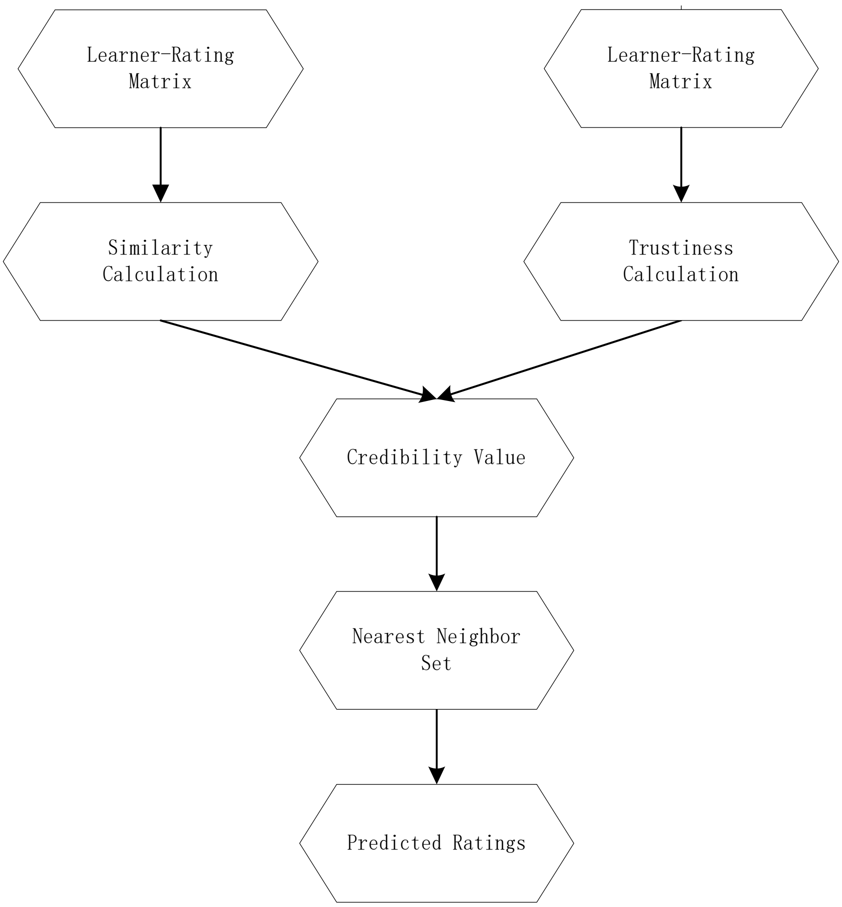

Figure 3. Flowchart of the proposed learner-based CF algorithm

For learners Alice and Bob, this paper articulates trustiness between them using the product of the ratio of common-rated OLRs to all OLRs rated by learner Alice and the ratio of common-rated OLRs to the total OLRs rated by both learner Alice and Bobs:

$$I_{a,b} = \frac{|I_a \cap I_b|}{|I_a|} * \frac{|I_a \cap I_b|}{|I_a \cup I_b|} \tag{5}$$

where I_a and I_b denote the sets of OLRs rated by learner Alice and Bob, respectively. The trust metric considers not only the proportion of common-rated OLRs in learner Alice's set, but also the proportion of common-rated OLRs in both learners' sets.

In the calculation of learner similarity, the first step involves computing the learner similarity *sim(a, b)* based on the modified Pearson coefficient (Equation 2). To address the issue of popular OLRs not highlighting personalization in original CF algorithm, a dynamic time weight for learner Alice's rating of OLR *i* is introduced, as follows:

$$w_{at(i)} = \frac{1}{\lg(1 + N_{t_u(i)})} \tag{6}$$

where $i \in I_{ab}$. Let $t_{\min\ a}$ represent the learner's first rating time for the OLR, and $t_{\max\ a}$ denote the learner's most recent rating time for the OLR.
The variable t_a signifies the learner's rating time period, specifically $t_{\min\ a} < t_a < t_{\max\ a}$. $N_{t_u(i)}$ indicates the number of times OLR *i* has been rated within t_a.

Next, calculate the temporal weight for each OLR based on the frequency of ratings within the time period. Finally, incorporate the time weight into the similarity calculation. Introducing dynamic time weight for learner similarity calculation, as follows:

$$sim_{at}(a,b) = \frac{\sum_{i \in I_{ab}} (r_{ai} - \bar{r}_a)(r_{bi} - \bar{r}_b) w_{at(i)}}{\sqrt{\sum_{i \in I_a} (r_{ai} - \bar{r}_a)^2 w_{at(i)}} \sqrt{\sum_{i \in I_b} (r_{bi} - \bar{r}_b)^2 w_{at(i)}}} \tag{7}$$

Utilizing similarity and trustiness as weights, we can obtain the credibility value $Cred_{a,b}$:

$$Cred_{a,b} = sim(a,b) * \tau(a,b) \tag{8}$$

where $Cred_{a,b}$ represents the credibility value between learner Alice and Bob. $sim_{a,\ b}$ denotes the similarity between learner Alice and Bob. $\tau(\ ,\ a\ b\ )$
signifies the trustworthiness between learner Alice and Bob. Selecting the top *k* learners whom learner Alice trusts the most, the neighbor set $Trust_{a,k}$ of learner Alice is formed. Using the credibility values $Cred_{a,b}$ from Alice to the learners in the nearest

neighbor set as weights, we calculate the predicted rating $p_1(r_{a,i})$ for Alice's unknown OLR i:

$$p_1(r_{a,i}) = \bar{r}_a + \frac{\sum_{b\in N(i)\cap Trust_{a,k}} Cred_{a,b}(r_{bi} - \bar{r}_b)}{\sum_{b\in N(i)\cap Trust_{a,k}} Cred_{a,b}} \quad (9)$$

where $\bar{r}_a$ and $\bar{r}_b$ denote the average ratings given by learners Alice and Bob to OLRs, respectively. $N(i)$ represents the set of learners who have rated OLR i. $Trust_{a,k}$ represents the set of k learners most trusted by Alice. r_{bi} is the rating given by learner Bob to OLR i. $Cred_{a,b}$ indicates the credibility value between learners Alice and Bob.

4.2. Introducing Dynamic Time Weight for OLR Similarity Calculation

Traditional item-based CF calculates similarity based on the assumption that learners might be interested in OLRs similar to their historical preferences. It identifies similar OLRs by analyzing learners' historical evaluation data and recommends based on the similarity of these OLRs. However, it neglects the reliability of learner ratings. Assuming learner Alice may casually rate an OLR within a certain period due to environmental or time-related factors, the score at that time is unreliable and can affect similarity calculations. Therefore, we introduce $w_{it(a)}$ to represent the dynamic time weight of OLR i:

$$w_{it(a)} = \frac{1}{\lg(1 + N_{t_i(a)})} \quad (10)$$

where $a \in A_{ij}$. Let $t_{\min i}$ denotes the first time a learner rated the OLR i. Let $t_{\max i}$ be the most recent time the OLR was rated, and t_i represents the time period during which the OLR was rated, i.e., $t_{\min i} < t_i < t_{\max i}$. $N_{t_i(a)}$ indicates the number of OLRs learner Alice rated within t_i.

After introducing dynamic time weight for OLRs, the calculation of OLR similarity is expressed as:

$$sim_{it}(i,j) = \frac{\sum_{a\in A_{ij}} (r_{ai} - \bar{r}_i)(r_{aj} - \bar{r}_j) w_{it(a)}}{\sqrt{\sum_{a\in A_{ij}} (r_{ai} - \bar{r}_i)^2 w_{it(a)}} \sqrt{\sum_{a\in A_{ab}} (r_{aj} - \bar{r}_j)^2 w_{it(a)}}} \quad (11)$$

By calculating OLR similarity, obtain a similarity matrix, and perform a descending sort on the similarities. Select the top k similar OLRs to form a set of similar OLRs, denoted by set $M(i)$. Based on the selected set of similar OLRs, the predicted rating $p_2(r_{ai})$ for learner Alice and OLR i is calculated:

$$p_2(r_{ai}) = r_i + \frac{\sum_{j\in M(i)} sim_{it}(i,j)(r_{aj} - \bar{r}_j)}{\sum_{j\in M(i)} sim_{it}(i,j)} \quad (12)$$

4.3. Improved Hybrid CF Algorithm

Due to the lower recommendation quality obtained through rating predictions based on either OLR or learner preferences, this paper chooses to utilize a hybrid CF based on dynamic time weights:

$$p(r_{ai}) = (1-\theta)(p_1(r_{ai})) + \theta(p_2(r_{ai})) \quad (13)$$

Where θÎ [0,1] is a tuning factor representing the dependency on p_1 and p_2. When $\theta = 0$, the algorithm considers only learner information. When $\theta = 1$, it considers only OLR information. Taking intermediate values implies a comprehensive consideration of both sources of information.

4.4. Cold Start Issue

In learner's perspective, CS is the issue of recommending suitable OLRs for new learners, including newly registered students, newly hired teachers, etc. To address learner CS issue, similarity between learners can be calculated based on natural attributes such as gender, age, grade, title, and college/unit. OLRs historically learned by learners with high similarity are recommended to the target learner. The method involves extracting learner-related attributes as vector features by using one-hot encoding, using 0 and 1 to represent discrete attributes like gender, and employing Min-Max normalization for continuous attributes within rage [0, 1]. For recommending existing OLRs to new learners, the proposed RS is applied to predict rating information, selecting the TOP-*N* exiting OLRs for recommendation. When recommending new OLRs to new learners, firstly obtain the collection of new OLRs recommended to old learners, perform deduplication, and then provide the list to the new learners.

The simplest way to handle OLR CS issue is to randomly showcase new OLRs, but this lacks personalization, and there is a high probability that the showcased new OLRs are not preferred by learners. Using CB (Content-Based) algorithm can solve this issue [26]. The specific method involves constructing a feature vector for new OLRs, extracting user preference feature vectors, calculating the similarity between them, and recommending new OLRs with high similarity to the target learner. Assuming the reader's preference vector is $A = (a_1, a_2, ..., a_n)$, with the corresponding preference weight vector $A' = (a'_1, a'_2, ..., a'_n)$, where the feature a'_i indicates the proportion of the number of OLRs of a certain type in all types of OLRs, ranging from 0 to 1. The feature vector of OLRs is $I = (i_1, i_2, ..., i_n)$, with features indicating whether the OLR belongs to a certain type, where 0 means no and 1 means yes.

The reader's preference weight is assigned, resulting in the weighted OLR vector I':

$$I' = (i'_1, i'_2, ..., i'_n) = (i_1 \times a'_1, i_2 \times a'_2, ..., i_n \times a'_n) \quad (14)$$

Next, the cosine similarity is used to calculate the similarity between the learner's preference and the OLR:

$$\cos(A, I) = \frac{\sum A'_i \times I'_i}{\sqrt{\sum A'^2_i \times \sum I'_i}} \quad (15)$$

Finally, the similarity is sorted to form a TOP-N recommendation of new OLRs with higher similarity for the learners.

5. Experiment

The experiment is conducted on an Intel(R) Core(TM) i5-12400 CPU environment using TensorFlow as the backend and Python language for compilation. Mean absolute error (MAE) and root mean square error (RMSE) are used as evaluation metrics. The experiment utilizes the Amazon 5-core Book dataset [27], consisting of 239,282 learners, 170,759 books, 12,278,677 ratings, and a density of 0.03%. Despite the large number of learners and OLRs, the rating behavior is sparse, indicating the dataset's sparsity. To address this, preprocessing is applied to ensure each learner and OLR have at least 5 rating instances. The experiment extracts partial features from these datasets, including learner ID, OLR ID, and learner ratings for OLRs (1-5 points). For the experiment, 80% of learner rating data is used as the training set, while the remaining 20% is used as the testing set.

5.1. Evaluation Metrics

MAE calculates the average difference between actual and predicted values, indicating the proximity of predictions to real values. RMSE is obtained by taking the square root of the ratio of the sum of squared differences between predicted and actual values to the total count [28]. Due to RMSE's sensitivity to prediction fluctuations, it effectively verifies the stability of different models:

$$MAE = \frac{\sum_{i=1}^{N} | r_{ai} - \hat{r}_{ai} |}{N} \quad (16)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (r_{ai} - \hat{r}_{ai})^2}{N}} \quad (17)$$

where $\hat{r}_{ai}$ and r_{ai} represent the predicted and actual ratings of user u for item i, respectively.

N denotes the number of ratings in the test set. Lower MAE and RMSE values indicate closer alignment between predicted and actual results, reflecting higher algorithm precision.

5.2. Experimental Results

Due to the significant impact of the number of neighbors on the accuracy of predicted ratings, the experiment compared the optimal values of parameter θ within the range [0, 1] at intervals of 0.2. Parameter θ in Equation (13) was introduced into the proposed CF algorithm to evaluate the algorithm's dependence on dynamically weighted OLRs and learner factors. The MAE results are depicted in Figure 4. It can be observed that when $\theta = 0$, the MAE value depends on the learner CF algorithm. When θ is between 0 and 1, the prediction combines the advantages of learner and OLR CF algorithms, effectively enhancing prediction accuracy. As θ increases, the prediction leans toward OLR-based aspects, and MAE gradually increases, indicating an increase in prediction error. When $\theta = 1$, the MAE value depends on the OLR CF algorithm. In summary, relying solely on learner or OLR CF algorithm at both ends yields lower recommendation accuracy. Optimal prediction accuracy is achieved when combining the strengths of both algorithms at $\theta = 0.4$.

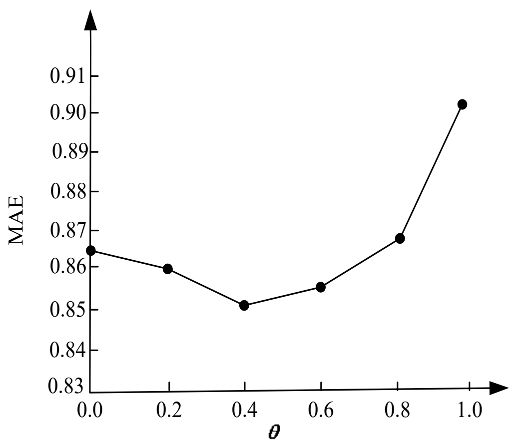

Figure 4. MAE result under different θ

To validate the superiority and effectiveness of the proposed improved algorithm, ablation study was carried out with different module compositions, as shown in Table 1, in which K donates the number of neighbors in CF algorithm. In the table, Model 1 employs only the learner CF algorithm. Model 2 exclusively utilizes the OLR CF algorithm. Model 3 employs a mixed algorithm, where θ is set to 0.4. Model 4 incorporates dynamic time weighting on top of this. Model 5 additionally introduces the calculation of credibility values.

Model 6 further integrates the CS solution, representing the complete proposed model. The results indicate that as K increases, the MAE and RMSE values gradually decrease. The introduction of dynamic time weighting improves prediction accuracy, highlighting the positive role of dynamic time weighting in model prediction accuracy.

Additionally, the hybrid algorithm, which combines both learner and OLR algorithms, outperforms single algorithms that only consider the impact of OLRs or learners unilaterally, resulting in low prediction accuracy. Overall, the MAE and RMSE values of the Model 6 are significantly lower than that of other models, indicating that the predicted values are closer to the actual values.

Table 1. Ablation study

		K = 5	K = 10	K = 15	K = 20	K = 25	K = 30
Model 1	MAE	1.157	0.998	0.947	0.893	0.888	0.884
	RMSE	1.305	1.292	1.248	1.231	1.176	1.134
Model 2	MAE	1.207	1.152	1.109	1.077	0.923	0.912
	RMSE	1.388	1.365	1.343	1.319	1.284	1.167
Model 3	MAE	1.071	0.933	0.894	0.880	0.873	0.875
	RMSE	1.249	1.236	1.231	1.186	1.153	1.157
Model 4	MAE	0.992	0.925	0.890	0.872	0.867	0.870
	RMSE	1.179	1.182	1.187	1.186	1.147	1.157
Model 5	MAE	0.874	0.871	0.868	0.864	0.860	0.862
	RMSE	1.149	1.152	1.151	1.150	1.097	1.153
Model 6	MAE	0.853	0.857	0.854	0.852	0.851	0.853
	RMSE	1.082	1.087	1.085	1.083	1.081	1.083

Figure 5 and 6 present performance comparison of the proposed method with other recently proposed RSs. The methods in [15] and [16] did not consider time factors. The approach in [19] did not combine the hybrid algorithm with time factors, limiting its ability to effectively utilize user and project information. The method in [21] used a deep learning approach, which may face challenges in the sparse data scenario of educational resource recommendations, potentially leading to overfitting and affecting model performance. The algorithm proposed in this paper achieved the best results. This is because the proposed method introduces a trust model by analyzing user interaction and rating behaviors, better reflects the level of user endorsement for each other, thereby improving recommendation accuracy. The dynamic time weighting factor is introduced into hybrid CF algorithm, considers not only the similarity between learners and OLRs, but also how these similarities change over time. This is crucial for capturing changes in learner preferences and behaviors, especially in online learning environments where learners' interests may evolve over time. By integrating credibility value between learners, the proposed algorithm comprehensively considers relationships between learners rather than solely relying on similarity. In addition, CF and CB algorithms are merged to address the CS issue. Overall, the proposed algorithm's innovation in modeling trustiness relationships between learners, considering dynamic time weighting, and integrating credibility values makes it more accurate in capturing changes in learner preferences and behaviors, thereby enhancing recommendation performance.

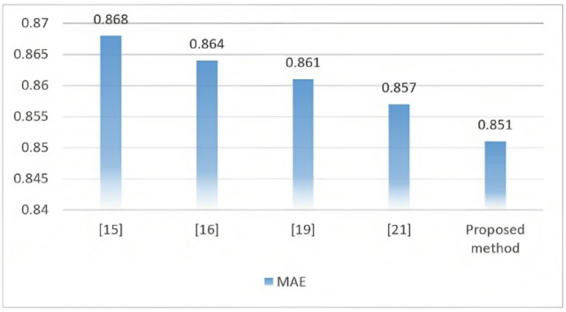

Figure 5. MAE results of different algorithms

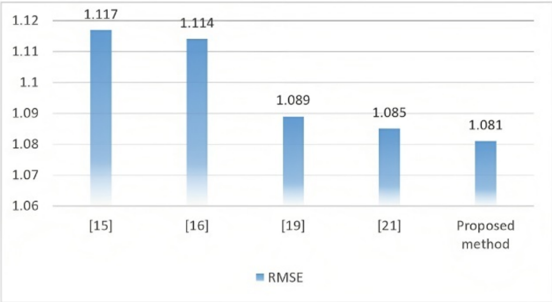

Figure 6. RMSE results of different algorithms

6. Conclusion

To effectively enhance the recommendation quality under online teaching environment, and address issues in traditional CF algorithms, such as the impact of popular OLRs, and an inability to identify learner-OLR preferences over time, we propose a hybrid CF recommendation algorithm based on dynamic time weighting and learner credibility. Initially, we analyze learner interaction and rating behaviors to build trustiness between users. Then learner similarity and trustiness are merged to calculate learner credibility values. Finally, the ratings of unknown learner OLRs are calculated. Subsequently, dynamic time weighting is added to both OLR-based and learner-based algorithms. The experiments show that in dealing with the sparse learner rating data, the hybrid credibility calculation significantly improves the accuracy of the model's recommendation predictions, demonstrating the superiority of the proposed algorithm. Considering the dynamic changes in learner interests and the evolving trust relationships among learners, in the future, we plan to explore deep reinforcement learning methods. This entails receiving rewards (ratings) from the teaching scenarios based on learner-OLR interactions, and updating model parameters.

References:

[1]. Mahdi, M. N., Ahmad, A. R., Ismail, R., Subhi, M. A., Abdulrazzaq, M. M., & Qassim, Q. S. (2020). Information Overload: The Effects of Large Amounts of Information. In *2020 1st. Information Technology to Enhance E-learning and Other Application (IT-ELA)*, 154–159. Baghdad, Iraq. Doi: 10.1109/it-ela50150.2020.9253082

[2]. Chen, J., Dong, H., Wang, X., Feng, F., Wang, M., & He, X. (2023). Bias and debias in Recommender System: A survey and future directions. *ACM Transactions on Information Systems, 41*(3), 1–39. Doi: 10.1145/3564284

[3]. Moore, R. L., & Blackmon, S. J. (2022). From the learner's perspective: A systematic review of MOOC learner experiences (2008–2021). *Computers & Education, 190*, 104596. Doi: 10.1016/j.compedu.2022.104596

[4]. Zulkifli, N., Hamzah, M. I., & Bashah, N. H. (2020). Challenges to teaching and learning using MOOC. *Creative Education, 11*(03), 197–205. Doi: 10.4236/ce.2020.113014

[5]. Pérez-Almaguer, Y., Toledo, R. Y., Alzahrani, A. A., & Martínez, L. (2021). Content-based group recommender systems: A general taxonomy and further improvements. *Expert Systems With Applications, 184*, 115444. Doi: 10.1016/j.eswa.2021.115444

[6]. Singh, P. K., Pramanik, P. K. D., & Choudhury, P. (2020). Collaborative filtering in recommender systems: technicalities, challenges, applications, and research trends. In *Apple Academic Press eBooks*, 183–215. Doi: 10.1201/9781003007210-8

[7]. Bai, M. L., Pamula, R., & Jain, P. K. (2019). Tourist Recommender System using Hybrid Filtering. In *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*, 746–749. Mathura, India. Doi: 10.1109/iscon47742.2019.9036308

[8]. Jain, A., Nagar, S., Singh, P. K., & Dhar, J. (2020). EMUCF: Enhanced multistage user-based collaborative filtering through non-linear similarity for recommendation systems. *Expert Systems With Applications, 161*, 113724. Doi: 10.1016/j.eswa.2020.113724

[9]. Singh, P. K., Sinha, M., Das, S., & Choudhury, P. (2020). Enhancing recommendation accuracy of item-based collaborative filtering using Bhattacharyya coefficient and most similar item. *Applied Intelligence, 50*(12), 4708–4731. Doi: 10.1007/s10489-020-01775-4

[10]. Jiang, L., Cheng, Y., Yang, L., Li, J., Yan, H., & Wang, X. (2018). A trust-based collaborative filtering algorithm for E-commerce recommendation system. *Journal of Ambient Intelligence and Humanized Computing, 10*(8), 3023–3034. Doi: 10.1007/s12652-018-0928-7

[11]. Duricic, T., Lacic, E., Kowald, D., & Lex, E. (2018, September). Trust-based collaborative filtering: Tackling the cold start problem using regular equivalence. In *Proceedings of the 12th ACM conference on recommender systems*, 446-450.

[12]. Bellogín, A., Castells, P., & Cantador, I. (2013, May). Improving memory-based collaborative filtering by neighbour selection based on user preference overlap. In *Proceedings of the 10th conference on open research areas in information retrieval*, 145-148.

[13]. Sachdeva, N., Wu, C. J., & McAuley, J. (2021). Svp-cf: Selection via proxy for collaborative filtering data. *arXiv preprint arXiv:2107.04984*.

[14]. De Sena Rosa, R. E. V., Guimarães, F. a. S., Da Silva Mendonça, R., & De Lucena, V. F. (2020). Improving prediction accuracy in Neighborhood-Based collaborative filtering by using local similarity. *IEEE Access, 8*, 142795–142809. Doi: 10.1109/access.2020.3013733

[15]. Ortega, F., Rojo, D., Valdiviezo, P., & Raya, L. (2018). Hybrid collaborative filtering based on users rating behavior. *IEEE Access, 6*, 69582–69591. Doi: 10.1109/access.2018.2881074

[16]. Zhang, C., Yang, M., Lv, J., & Yang, W. (2018). An improved hybrid collaborative filtering algorithm based on tags and time factor. *Big Data Mining and Analytics, 1*(2), 128–136. Doi: 10.26599/bdma.2018.9020012

[17]. Liao, X., Wu, H., & Wang, Y. (2020). ANT Collaborative filtering addressing sparsity and temporal effects. *IEEE Access, 8*, 32783–32791. Doi: 10.1109/access.2020.2973931

[18]. Wangwatcharakul, C., & Wongthanavasu, S. (2020). Dynamic collaborative filtering based on user preference drift and topic evolution. *IEEE Access, 8*, 86433–86447. Doi: 10.1109/access.2020.2993289

[19]. Joorabloo, N., Jalili, M., & Ren, Y. (2020). Improved collaborative filtering recommendation through similarity prediction. *IEEE Access, 8*, 202122–202132. Doi: 10.1109/access.2020.3035703

[20]. Hou, Y., Zhou, P., Wang, T., Yu, L., Hu, Y., & Wu, D. (2016). Context-aware online learning for course recommendation of MOOC big data. *arXiv preprint arXiv:1610.03147*.

[21]. Vedavathi, N., & Kumar, K. (2022). SentiWordNet Ontology and deep neural network based collaborative filtering technique for course recommendation in an E-Learning platform. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 30*(4), 709–732. Doi: 10.1142/s0218488522500192

[22]. Ng, Y., & Linn, J. (2017). CrsRecs: A personalized course recommendation system for college students. In *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)*, 1–6. Larnaca, Cyprus. Doi: 10.1109/iisa.2017.8316368

[23]. Rani, L., Wise, D. J. W., Ajayram, K., Gokul, T., & Kirubakaran, B. (2020). Course Recommendation for students using Machine Learning. In *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC).* Doi: 10.1109/icesc48915.2020.9156012

[24]. Pan, Z., Zhao, L., Zhong, X., & Xia, Z. (2021). Application of Collaborative Filtering Recommendation Algorithm in Internet Online Courses. In *ICBDC '21: Proceedings of the 6th International Conference on Big Data and Computing*, 142–147. Doi: 10.1145/3469968.3469992

[25]. Jin, W. (2023). User interest modeling and collaborative filtering algorithms application in English personalized learning resource recommendation. *Soft Computing.* Doi: 10.1007/s00500-023-08700-0

[26]. Javed, U., Shaukat, K., Hameed, I. A., Iqbal, F., Alam, T. M., & Luo, S. (2021). A review of Content-Based and Context-Based recommendation Systems. *International Journal of Emerging Technologies in Learning (Ijet), 16*(3), 274. Doi: 10.3991/ijet.v16i03.18851

[27]. AlZu'bi, S., Alsmadi, A., Al-Qatawneh, S. M., Al-Ayyoub, M., Hawashin, B., & Jararweh, Y. (2019). A Brief Analysis of Amazon Online Reviews. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS),* 555–560. Granada, Spain. Doi: 10.1109/snams.2019.8931816

[28]. Nafea, S. M., Siewe, F., & He, Y. (2019). On recommendation of learning objects using Felder-Silverman Learning Style model. *IEEE Access, 7*, 163034–163048. Doi: 10.1109/access.2019.2935417