Vietnam Journal of Science and Technology ***62*** *(4) (2024) 1005-1017*
doi:10.15625/2525-2518/18961

ADCMDES: Design of an augmented cross-domain collaborative recommendation model using novel distance metric and ensemble stratification

Indu Hariyale[1, *], Mukesh. M. Raghuwanshi[2]

[1] *Yashwantrao Chavan College of Engineering Hingna Rd, Wanadongri ct, Nagpur, Wanadongri, Maharashtra, India, Nagpur, India,*

[2] *Computer Engineering, Symbiosis Institute of Technology, Nagpur (SIT Nagpur), Symbiosis International (Deemed University) (SIU), Wathoda, Nagpur, Maharashtra, India,*

[*] Emails: *induhrl@gmail.com, director@sitnagpur.siu.edu.in*

Received: 20 September 2023; Accepted for publication: 18 November 2023

Abstract. Collaborative recommendation systems utilize data from one entity to predict features of other entities through pattern analysis, crucial for understanding data behavior impacts. Researchers use various models and distance metrics, like Jaccard and Cosine, to determine correlations between user queries and recommendation datasets. However, these models face efficiency challenges as datasets grow, causing delays in correlation estimation. To address this, the ADCMDES model was developed. This advanced, semi-supervised model enhances scalability by using a hybrid distance metric and ensemble stratification for dataset pruning. It clusters similar entities, using word2vec to convert records into features for an ensemble classification engine. The model directs user queries to the most relevant cluster, ranking entries with a hybrid metric combining 18 distance measures. The ADCMDES has shown improvements in accuracy (15 %), precision (8 %), recall (9 %), and a 3% reduction in RMSE compared to traditional models. Although it introduces some delays, these can be mitigated with parallel processing.

Keywords: Recommendation, hybrid, ensemble, augmented, distance, correlation, stratification.

Classification numbers: 4.7.4., 4.8.3., 4.8.4.

1. INTRODUCTION

The article addresses the challenge of inefficiency in collaborative recommendation models, particularly as dataset sizes grow, causing delays in correlation estimation. The solution involves enhancing scalability without compromising recommendation quality. The key research contribution is ADCMDES (Augmented Cross-Domain Collaborative Model using a Novel Distance Metric and Ensemble Stratification). ADCMDES improves scalability by using a hybrid distance metric and ensemble stratification for dataset pruning and operates semi-supervisedly with information on the entities being analyzed. It employs word2vec to convert records into features for an ensemble classification engine, which categorizes user inputs and directs them to the most relevant cluster. In testing, the ADCMDES demonstrated superior accuracy, precision, recall, and lower RMSE compared to standard models, proving its effectiveness. A typical cross-domain recommendation engine is depicted in Figure 1, wherein user ratings about products are linked with their buying patterns in order to facilitate better product recommendations [1].

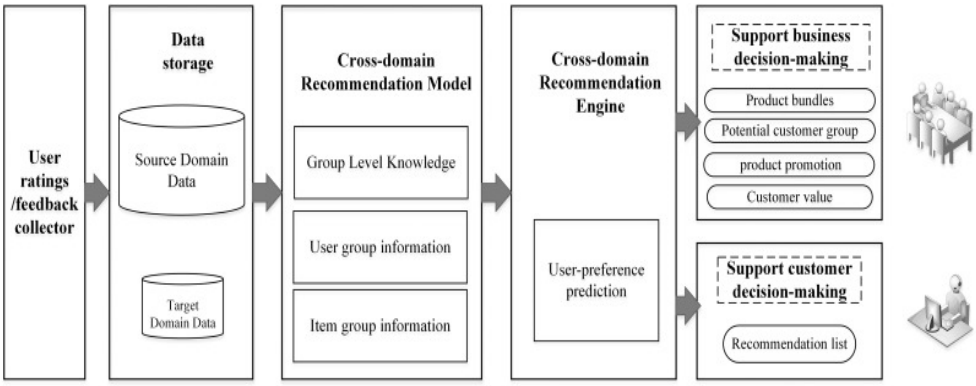

Figure 1. A typical cross-domain recommender model.

The model initially processes user ratings, and converts them into group-level knowledge via user-level and product-level grouping, which assists in identification of similar rating products by similar type of users. These groups are given to user-preference prediction engine, which produces product recommendations, depending upon temporal user-preferences. These preferences are matched with other users in order to estimate a correlative product-to-user mapping metric. This metric is evaluated via equation (1) as follows,

$$C_{UP} = \frac{\sum_{i=1}^{N_p} \left(M_u - M_p\right)^2}{\sqrt{\sum_{i=1}^{N_p} M_u^2 - M_p^2}} \tag{1}$$

here, $C_{up}, M_u, M_p,$ and N_p represent correlation score, user matching score, product matching score, and number of products used for recommendation, respectively. Various models for optimizing correlation values and enhancing recommendation performance are reviewed. The next section introduces the augmented cross-domain collaborative recommendation model, featuring a novel distance metric and ensemble stratification.

2. LITERATURE REVIEW

Researchers have proposed numerous cross-domain recommendation (CDR) models over the years, each differing in application, performance, and recommendation metrics [1]. For example, works in [2 - 4] introduce models like multiple source CDR, Preference Structure Information Sharing (PSIS) CDR, and reference-based CDR, leveraging feature variance for specific recommendations. These were extended in [5, 6] with Transition-based CDR and Graph Analysis with user-item embedding, improving efficiency with minimal delay but limited to specific contexts. To enhance scalability, the Multiple Domain Semantic Fusion Model (MDSFM) [7] uses collaborative filtering but can be improved through deep and transfer learning, as explored in [8-10], which introduced deep feature learning methods. These approaches include models like Deep Collaborative Filtering (DCF) with Geometric Structure Preservation [11], deep neural networks [12], autoencoders [13], and variational autoencoders [15], all boosting recommendation accuracy. Linear recommendation models in [16 - 18] discuss collaborative ranking and correlation-based matching but have moderate accuracy due to limited feature extraction. Performance is improved in [19-21] through Kernel-Induced Knowledge Transfer (KIKT) and adversarial learning, while a review in [22] outlines content-based, collaborative, and hybrid filters. From this, models like public opinion-based, hybrid, and context-based recommendations have been proposed to further enhance recommendation efficiency across various domains.

3. PROPOSED AUGMENTED CROSS-DOMAIN COLLABORATIVE RECOMMENDATION MODEL USING NOVEL DISTANCE METRIC AND ENSEMBLE STRATIFICATION

From the literature review it is observed that a wide variety of machine learning models are available for cross domain recommendation, and these models are used for application-specific recommendation scenarios. Moreover, these models utilize distance metrics, which are highly-optimized for context-sensitive recommendations, and thus cannot be scaled to larger variety of datasets. The proposed model also uses an ensemble stratification engine, which combines random forest (RF), logistic regression (LR), linear support vector machine (LSVM), and multilayer perceptron (MLP) classifiers identification of common cross-domain metrics. These metrics are combined in order to generate final cross-domain recommendations. Overall flow of the model is described in Figure 2, wherein different recommendation and collaboration blocks are combined to form the final recommendation engine.

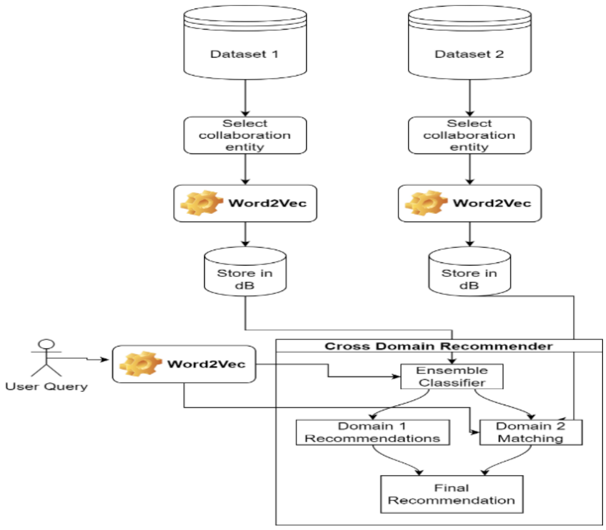

Figure 2. Design of the proposed recommendation model.

From Figure 2, it is observed that a semi-supervised layer requires user inputs in the form of collaboration entity selection, which is followed by entity-level word2vec based feature extraction. These features are stored in different databases, and used by different blocks for cross-domain recommendation. The features of source dataset are compared with user query, and single-domain recommendations are formed. Features extracted via this recommendation are used for matching target domain entities, and target domain results are recommended to the user. These results are evaluated using an augmented distance metric, while classification is performed using an ensemble classification engine. Design of these engines is described in different sub-sections of this text.

3.1. Design of feature extraction with ensemble classification

Both source and target datasets are input to the system, and their collaboration entities are selected. Each of these are uniquified, and a per-entity feature set is extracted via equation (2),

$$FS_{E_i} = \bigcup_{j=1}^{NE} F_{E_i,j} \tag{2}$$

Here, FS_{E_i} represents unified feature set for Entity E_i, NE represents number of unique entities, and $F_{E_i,j}$ represents current feature set of the entity. The word2vec model uses probability maximization in order to estimate logarithmic feature set as described in equation (3),

$$L_f = \frac{1}{T} * \log\left(\sum_{t=1}^{T}\sum_{j=-w}^{w} P\left(W_{t+j}, W_t\right)\right) \tag{3}$$

Here, L_f represents logarithmic probability, T represents position of the word, W_t represents word at the position, w represents window size, and $P(W_1, W_2)$ represents probability of occurrence of word W_1 with W_2, which is represented via equation (4), wherein exponential vectors are used for estimation of word-to-word mapping,

$$P(W_1, W_2) = \frac{\exp\left(\frac{C(W_1,W_2)}{C(W_1,W_2)+C(W_2,W_1)}\right)}{\sum_{i=1}^{N} \exp\left(\frac{C(W_1,W_i)}{C(W_1,W_i)+C(W_i,W_1)}\right)} \tag{4}$$

Here, $C(X, Y)$ represents conditional probability of occurrence of word X before word Y in the entire corpus, while N represents total number of words in the corpus. Finally, these values are combined to get the word2vec features via equation (5) as follows,

$$F_{w2v} = \frac{\sum_{i=1}^{N_c}\sum_{j=1}^{N_w} P(W_i, W_j)}{N_c * N_w} \tag{5}$$

Here, F_{w2v} represents the final word2vec features, N_c and N_w denote the number of words in the corpus and input document, respectively, while P is the word corpus probability. Gensim and BERT corpus models are used to train the system and generate large-scale feature vectors. To achieve this, a novel variance threshold is calculated using equation(5).

$$V_{th} = \sqrt{\frac{\sum_{a=1}^{m}\left(x_a - \frac{\sum_{i=1}^{m}\sqrt{\frac{\sum_{j=1}^{n}\left(x_j - \frac{\sum_{k=1}^{n} x_k}{n}\right)^2}{n-1}}}{m}\right)^2}{m-1}} \tag{6}$$

Where, x represents extracted word2vec features, n and m represents total number of features extracted for current group of words, and total number of features extracted for other group of words. Word groupings are evaluated via correlation maximization, wherein equation (7) is used to evaluate feature matching between different words as follows,

$$C_{W_1,W_2} = \frac{\sum_{i=1}^{N_f}\left(F_{i_{W_1}} - F_{i_{W_2}}\right)^2}{\sqrt{\sum_{i=1}^{N_f} F_{i_{W_1}}^2 - F_{i_{W_2}}^2}} \tag{7}$$

Where, N_f represents number of extracted features, F_i represents the feature vector, and $C_{1,2}$ represents correlation between the 2 feature vectors. Vectors with matching values of correlation are clubbed together to form groups, which are used to find threshold variance. Features with variance lower than V_{th} are removed from the vector, while others are selected and given to the ensemble classifier. The ensemble classifier uses combination of random forest (RF), logistic regression (LR), linear support vector machine (LSVM), and multilayer perceptron (MLP) in order to obtain the final category of recommendation. Design of the proposed ensemble classifier can be observed from Figure 3.

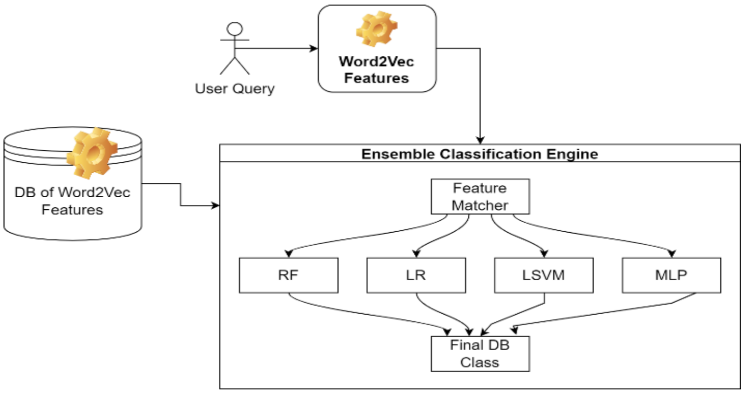

Figure 3. Ensemble model for source domain recommendation.

The word2vec features are given to each of the classification engines, which use standard models to identify the best matching type for current input. All records from this type are given to the final recommendation engine in order to perform cross-domain recommendation. Design of proposed novel metric-based engine is described in the next section of this text.

3.2. Design of the novel metric for final recommendation

In order to improve efficiency of recommendation, word2vec features of the results from source domain are matched with word2vec features of all items in the target domain. This matching is performed using a novel recommender metric, which is evaluated via equation (8) as follows,

$$NM(i,j) = \frac{1}{18} * \big(BaC(i,j) + Can(i,j) + Cheb(i,j) + CBlock(i,j) + Corr(i,j) + Cos(i,j) + Euc(i,j) + M(i,j) + Euc^2(i,j) + Dice(i,j) + Hamming(i,j) + Jac(i,j) + Kul(i,j) + Roger(i,j) + RR(i,j) + SM(i,j) + SS(i,j) + Yule(i,j)\big) \tag{8}$$

where, $NM(i,j)$ represents the novel metric value between i^{th} feature vector of source domain, and j^{th} feature vector of destination domain, BaC indicates Bray Curtis feature Metric, Can indicates Canberra feature Metric, $Cheb$ indicates Chebyshev feature Metric, $CBlock$ indicates City block feature Metric, $Corr$ indicates Correlation feature Metric, Cos indicates Cosine distance feature Metric, Euc indicates Euclidean distance feature Metric, M indicates Minkowski feature Metric, $Dice$ represents dice coefficient feature Metric, $Hamming$ indicates Hamming feature Metric, Jac indicates Jaccard Distance feature Metric, Kul indicates Kulsinski feature Metric, $Roger$ indicates Rogerstani Moto feature Metric, RR indicates Russel Rao feature Metric, SM indicates Sokal Michener feature Metric, SS indicates Sokal Sneath feature Metric, and $Yule$ indicates Yule distance Metric values. Distances of each feature vector between source and target domain are identified, and a feature threshold is evaluated via equation (9) as follows,

$$F_{th} = \partial * \frac{\sum_{i=1}^{N_S} \sum_{j=1}^{N_T} NM(i,j)}{N_S * N_T} \quad (9)$$

where, ∂ is a tuning factor, and can be changed depending upon user preference. All records present in target entity, with novel metric more than F_{th} are recommended to the user. Based on user feedback related to accuracy, precision, recall, and RMSE, the value of tuning factor is modified via equation (10) as follows,

$$\partial = \partial_{old} + \left[\frac{A_{th} - A_{old}}{A_{old}} + \frac{P_{th} - P_{old}}{P_{old}} + \frac{R_{th} - R_{old}}{R_{old}}\right] * \left(\frac{RMSE_{old}}{RMSE_{th} - RMSE_{old} + 1}\right) \quad (10)$$

Where, $A, P, R, and\ RMSE$ represent values of accuracy, precision, recall, and RMSE for previous iteration, while x_{th} represents parametric threshold value which is needed for effective operation of the recommendation model.

4. RESULTS AND ANALYSIS

The ADCMDES model enhances cross-domain recommendations by combining ensemble classification with a novel recommendation metric. Its efficiency is assessed using RMSE and Precision. The RMSE evaluates rating prediction accuracy in collaborative filtering, with lower values indicating better performance. Precision measures the relevance of recommended items in binary settings, while recall gauges the proportion of relevant items recommended. The model's efficiency was tested using the following datasets:

- MovieLens dataset with Social Network influencer (https://grouplens.org/datasets/movielens/25m/, with https://www.kaggle.com/c/predict-who-is-more-influential-in-a-social-network/data)
- Books dataset with Million Songs dataset http://www2.informatik.unifreiburg.de/~cziegler/BX/, with http://millionsongdataset.com/)
- Amazon review with Yahoo Music user ratings (https://nijianmo.github.io/amazon/index.html
- withhttps://webscope.sandbox.yahoo.com/catalog.php)

Over 20,000 records from combined datasets were used for training and validation. User inputs were automated, and performance was assessed using a 70:15:15 split for training, testing, and validation. The following section details the experimental setup for the "ADCMDES" model, using the MovieLens, Books, Amazon reviews, and Yahoo Music datasets. Data was preprocessed by cleaning and merging, then split into training, testing, and validation sets. Experiments were conducted in a Python environment with libraries like NumPy, pandas, scikit-learn, TensorFlow, and PyTorch. Word2Vec was used for feature extraction from item content, user interactions, and metadata, with tools such as Gensim and BERT for embedding and feature vector creation.Evaluation Metrics: Functions are implemented to calculate evaluation metrics like accuracy, precision, recall, and RMSE as described in the article. Train and Test: Train the

ADCMDES model using the training dataset and evaluate its performance using the testing dataset. Experiment with different values of tuning parameters, such as ∂, to optimize the model's performance.

Table 1. Accuracy of recommendation for different number of records.

Num Recs.	$\boldsymbol{A_R}$ MS [2]	$\boldsymbol{A_R}$ DNN [12]	$\boldsymbol{A_R}$ [R3]	$\boldsymbol{A_R}$ Proposed
25	77.72	72.68	72.35	79.26
50	78.68	73.64	73.23	80.32
75	78.68	75.58	74.20	81.39
100	79.64	76.55	75.08	82.35
125	80.60	78.49	76.54	83.89
150	81.56	79.46	77.42	84.85
175	81.56	81.40	78.38	85.92
200	81.56	78.00	76.73	84.08
225	82.52	78.78	77.61	85.04
250	82.52	79.46	77.90	85.43
291	82.52	80.04	78.19	85.72
334	81.46	80.52	77.90	85.43
375	81.84	80.91	78.29	85.82
416	82.14	81.10	78.49	86.01
459	82.42	81.01	78.58	86.20
500	82.71	81.49	78.97	86.59
541	82.90	81.88	79.27	86.88
584	83.00	82.18	79.46	87.07
625	83.19	82.56	79.66	87.36
666	83.29	82.75	79.85	87.55
709	83.38	83.04	80.05	87.75
750	83.48	83.33	80.24	87.94
791	83.57	83.72	80.44	88.13
834	83.76	84.01	80.63	88.43
916	83.86	84.30	80.92	88.71
1000	84.06	84.69	81.11	88.90
1084	84.25	84.98	81.31	89.20
1166	84.34	85.27	81.61	89.39
1250	84.44	85.57	81.80	89.68
1334	84.63	85.95	81.99	89.87
1416	84.72	86.24	82.18	90.16
1500	84.92	86.54	82.48	90.35
1666	85.02	86.92	82.67	90.64
1834	85.21	87.21	82.87	90.83
2000	85.30	87.50	83.16	91.12

Table 2. Precision of recommendation for different validation records.

Num Recs.	$\boldsymbol{P_R}$ MS [2]	$\boldsymbol{P_R}$ DNN [12]	$\boldsymbol{P_R}$ [R3]	$\boldsymbol{P_R}$ Proposed
25	73.88	67.47	73.08	78.99
50	74.75	68.39	74.07	79.98
75	74.75	70.14	74.97	80.99
100	75.71	71.06	75.96	81.98
125	76.57	72.89	77.36	83.48
150	77.53	73.81	78.25	84.48
175	77.53	75.56	79.24	85.48
200	77.53	72.43	77.55	83.78
225	78.39	73.07	78.44	84.68
250	78.39	73.81	78.74	84.99
291	78.39	74.36	79.05	85.28
334	77.43	74.82	78.74	85.08
375	77.72	75.10	79.14	85.38
416	78.00	75.28	79.35	85.69
459	78.30	75.19	79.44	85.78
500	78.58	75.65	79.84	86.18
541	78.77	76.01	80.14	86.48
584	78.87	76.28	80.34	86.68
625	78.96	76.56	80.54	86.99
666	79.16	76.83	80.73	87.18
709	79.16	77.11	80.94	87.38
750	79.26	77.39	81.13	87.58
791	79.35	77.75	81.33	87.78
834	79.54	78.03	81.53	88.08
916	79.73	78.31	81.83	88.28
1000	79.83	78.58	82.02	88.58
1084	80.02	78.86	82.23	88.78
1166	80.12	79.13	82.42	88.98
1250	80.22	79.50	82.72	89.28
1334	80.41	79.78	82.93	89.48
1416	80.50	80.05	83.12	89.68
1500	80.60	80.33	83.32	89.98
1666	80.79	80.69	83.52	90.18
1834	80.88	80.96	83.82	90.48
2000	81.08	81.24	84.01	90.68

Compare with Other Models: Compare the performance of the ADCMDES model with other reference models mentioned in the article (MS, DNN, [R3]) for different recommendation applications. For instance, the test accuracy of recommendation (A_R) was estimated using the following equation (11),

$$A_R = \frac{R_C}{R_T} \tag{11}$$

where, R_C and R_T represents number of correct recommendations, and total number of recommendations used for evaluation. Accuracy values are tabulated in Table 1, wherein it is compared with standard cross-domain recommendation models on the same dataset. Based on this analysis it can be observed that the proposed model is 6 % more effective than MS, 3.5 % more effective than DNN, and 9 % more effective than [R3] for different recommendation applications. Using similar training and testing sets, precision of recommendation (P_R) was evaluated using equation (12),

$$P_R = \frac{R_{CI}}{R_T} \tag{12}$$

where, R_{CI} and R_T represent total number of correctly recommended entities with incorrect categories, and total number of recommended values used for validation. These results are tabulated in Table 2, wherein precision of the proposed model is compared with other reference models. Based on this analysis it can be observed that the proposed model is 9% more effective than MS, 8.5 % more effective than DNN, and 6% more effective than [R3] under different input types. Using similar training and testing sets, recall of recommendation (R_R) was evaluated using the following equation (13),

$$R_R = \frac{R_{CC}}{R_T} \tag{13}$$

where, R_{CC} and R_T represent number of correctly recommended results with correct classes, and total number of recommendation values used for testing. These results are tabulated in Table 3, wherein recall of the proposed model is compared with other reference models.

Table 3. Recall of classification for different test recommendation sets.

Num Recs.	$\boldsymbol{R_R}$ MS [2]	$\boldsymbol{R_R}$ DNN [12]	$\boldsymbol{R_R}$ [R3]	$\boldsymbol{R_R}$ Proposed
25	60.64	61.97	61.25	76.78
50	61.41	62.79	62.04	77.69
75	61.41	64.44	62.84	78.68
100	62.08	65.27	63.63	79.69
125	62.85	66.92	64.73	81.18
150	63.61	67.75	65.52	82.18
175	63.61	69.40	66.32	83.18
200	63.61	66.47	64.92	81.38
225	64.38	67.10	65.62	82.28
250	64.38	67.75	65.92	82.68
291	64.38	68.30	66.22	82.99
334	63.52	68.67	66.02	82.68
375	63.81	68.94	66.22	82.99
416	64.10	69.13	66.42	83.28

459	64.29	69.13	66.51	83.38
500	64.48	69.49	66.81	83.78
541	64.67	69.77	67.11	84.08
584	64.77	70.05	67.31	84.28
625	64.87	70.32	67.42	84.48
666	64.96	70.60	67.61	84.78
709	65.06	70.87	67.81	84.88
750	65.06	71.06	67.91	85.08
791	65.15	71.33	68.10	85.28
834	65.34	71.60	68.31	85.58
916	65.44	71.88	68.50	85.88
1000	65.53	72.15	68.71	86.08
1084	65.73	72.43	68.90	86.28
1166	65.83	72.70	69.10	86.58
1250	65.92	72.98	69.20	86.78
1334	66.02	73.26	69.40	86.99
1416	66.11	73.53	69.60	87.28
1500	66.21	73.81	69.80	87.48
1666	66.30	74.08	70.00	87.68
1834	66.40	74.36	70.20	87.98
2000	66.59	74.64	70.39	88.18

Based on this analysis it can be observed that the proposed model is 22 % more effective than MS, 14 % more effective than DNN, and 18 % more effective than [R3] under different cross-domain recommendation systems. Using similar training and testing sets, RMSE of recommendation was evaluated and tabulated in Table 4, and compared with other reference models.

Table 4. RMSE for different cross-domain recommendation systems.

Num Recs.	$RMSE$ MS [2]	$RMSE$ DNN [12]	$RMSE$ [R3]	$RMSE$ Proposed
25	6.66	6.46	6.66	5.19
50	6.74	6.55	6.75	5.25
75	6.74	6.72	6.84	5.32
100	6.82	6.80	6.92	5.39
125	6.91	6.98	7.05	5.49
150	6.99	7.06	7.14	5.55
175	6.99	7.23	7.22	5.62
200	6.99	6.93	7.07	5.51
225	7.07	7.00	7.15	5.57
250	7.07	7.07	7.18	5.59
291	7.07	7.11	7.21	5.61

334	6.99	7.16	7.18	5.59
375	7.01	7.19	7.21	5.61
416	7.03	7.21	7.23	5.63
459	7.06	7.21	7.25	5.64
500	7.08	7.24	7.28	5.67
541	7.10	7.28	7.30	5.69
584	7.11	7.31	7.32	5.70
625	7.12	7.34	7.34	5.71
666	7.14	7.36	7.36	5.73
709	7.14	7.38	7.38	5.74
750	7.15	7.41	7.39	5.75
791	7.16	7.44	7.41	5.77
834	7.17	7.47	7.44	5.79
916	7.19	7.50	7.46	5.81
1000	7.20	7.53	7.48	5.82
1084	7.22	7.56	7.50	5.83
1166	7.22	7.58	7.52	5.85
1250	7.23	7.61	7.54	5.87
1334	7.24	7.64	7.56	5.88
1416	7.26	7.67	7.58	5.90
1500	7.27	7.69	7.60	5.91
1666	7.28	7.72	7.62	5.93
1834	7.29	7.75	7.64	5.95
2000	7.31	7.79	7.66	5.96

The "ADCMDES: Augmented Cross-Domain Collaborative Recommendation Model" improves recommendation performance, being 15% more effective than MS, 18% more effective than DNN, and 16% more effective than [R3] in RMSE across various cross-domain scenarios. The model uses ensemble classification for precise source domain predictions and a novel metric for cross-domain checks, enhancing matching efficiency.

5. CONCLUSION AND FUTURE WORK

The proposed model initially combines unique feature vectors from source and target domains in order to generate a large-scale feature vector. This vector is reduced using a customized variance-based feature selection unit, which assists in reducing delay needed for classification and improving accuracy of single-domain recommendation. The reduced features are given to an ensemble classification model, which assists in recommending most probable single-domain entities based on user query. These entities are mapped with target domain using a novel distance metric, due to which an accuracy of 91.1 % across different domains is achieved. Furthermore, the proposed model is 6 % more effective than MS, 3.5 % more effective than DNN, and 9 % more effective than [R3] in terms of accuracy, while it is 15.4 % more effective than MS, 16.2 % more effective than DNN, and 9.6% more effective than [R3] in terms of average precision and recall measurements under different input types. Due to which its

RMSE is also reduced, and it is observed that the proposed model is 15 % more effective than MS, 18 % more effective than DNN, and 16 % more effective than [R3] in terms of RMSE for different cross-domain recommendation application scenarios. The model's performance can be further improved via use of Q-learning and incremental learning methods which assist in reducing redundancies, while improving recommendation performance for large-scale scenarios.

REFERENCES

1. Do Quan, Liu Wei, Fan Jin, Tao Dacheng - Unveiling Hidden Implicit Similarities for Cross-Domain Recommendation. IEEE Transactions on Knowledge and Data Engineering (2019) pp. 1-1. 10.1109/TKDE.2019.2923904.
2. Zhang Qian, Lu Jie, and Zhang Guangquan - Cross-Domain Recommendation with Multiple Sources (2020) 1-7. 10.1109/IJCNN48605.2020.9207014.
3. Zhu Nengjun and Cao Jian - Enhancing Cross-domain Recommendation through Preference Structure Information Sharing, (2020) 524-531. 10.1109/ICWS49710. 2020.00076.
4. Zawali Abir and Boukhris Imen - Cross Domain Collaborative Filtering Recommender System for Academic Venue Personalization based on References (2020) 2829-2835. 10.1109/SSCI47803.2020.9308377.
5. Doan T. N. and Sahebi S. - TransCrossCF: Transition-based Cross-Domain Collaborative Filtering, 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA) (2020) pp. 320-327, doi: 10.1109/ICMLA51294.2020.00059.
6. Hirakawa T., Maeda K., Ogawa T., Asamizu S., and Haseyama M. - Cross-domain Recommendation via Multi-layer Graph Analysis Using User-item Embedding, 2020 IEEE 9th Global Conference on Consumer Electronics (GCCE), 2020, pp. 714-715, doi: 10.1109/GCCE50665.2020.9292019.
7. Li Xiang, He Jingsha, Zhu Nafei, & Hou Ziqiang - Collaborative Filtering Recommendation Based on Multi-Domain Semantic Fusion, 2020, pp. 255-261. 10.1109/COMPSAC48688.2020.00041.
8. Singh P. K., Pramanik P. K. D., Mishra S., Nayyar A., Shukla D., and Choudhury P. - Improving Recommendation Accuracy using Cross-domain Similarity, 2020 7th NAFOSTED Conference on Information and Computer Science (NICS), 2020, pp. 280-285, doi: 10.1109/NICS51282.2020.9335913.
9. Liu Z., Tian J., Zhao L., and Zhang Y. - Attentive-Feature Transfer based on Mapping for Cross-domain Recommendation, 2020 International Conference on Data Mining Workshops (ICDMW), 2020, pp. 151-158, doi: 10.1109/ICDMW51313.2020.00030.
10. Kang Y., Gai S., Zhao F., Wang D., and Luo Y. - Cross-Domain Deep Collaborative Filtering for Recommendation, 2019 International Conference on Data Mining Workshops (ICDMW), 2019, pp. 634-638, doi: 10.1109/ICDMW.2019.00096.
11. Kang Y., Gai S., Zhao F., Wang D., and Tang A. - Deep Transfer Collaborative Filtering with Geometric Structure Preservation for Cross-Domain Recommendation, 2020 International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1-8, doi: 10.1109/IJCNN48605.2020.9207009.
12. Wang H. - A DNN-Based Cross-Domain Recommender System for Alleviating Cold-Start Problem in E-Commerce, in IEEE Open Journal of the Industrial Electronics Society, Vol. 1, pp. 194-206, 2020, doi: 10.1109/OJIES.2020.3012627.

13. Zhong S. T., Huang L., Wang C. D., Lai J. H., and Yu P. S. - An Autoencoder Framework With Attention Mechanism for Cross-Domain Recommendation, in IEEE Transactions on Cybernetics, doi: 10.1109/TCYB.2020.3029002.
14. Zhang Q., Hao P., Lu J. and Zhang G. - Cross-domain Recommendation with Semantic Correlation in Tagging Systems, 2019 International Joint Conference on Neural Networks (IJCNN), 2019, pp. 1-8, doi: 10.1109/IJCNN.2019.8852049.
15. Zhang Q., Hao P., Lu J., and Zhang G. - Cross-domain Recommendation with Semantic Correlation in Tagging Systems, 2019 International Joint Conference on Neural Networks (IJCNN), 2019, pp. 1-8, doi: 10.1109/IJCNN.2019.8852049.
16. Shi J. and Wang Q. - Cross-Domain Variational Autoencoder for Recommender Systems, 2019 IEEE 11th International Conference on Advanced Infocomm Technology (ICAIT), 2019, pp. 67-72, doi: 10.1109/ICAIT.2019.8935901.
17. Chen H. and Li J. - Collaborative Ranking Tags and Items via Cross-domain Recommendation, 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 721-730, doi: 10.1109/BigData47090.2019.9006062.
18. Yin J., Guo Y., and Chen Y. - Heterogenous Information Network Embedding Based Cross-Domain Recommendation System, 2019 International Conference on Data Mining Workshops (ICDMW), 2019, pp. 362-369, doi: 10.1109/ICDMW.2019.00060.
19. Qi Q., Cao J., Tan Y., and Xiao Q. - Cross-Domain Recommendation Method in Tourism, 2018 IEEE International Conference on Progress in Informatics and Computing (PIC), 2018, pp. 106-112, doi: 10.1109/PIC.2018.8706265.
20. Zhang Q., Lu J., Wu D., and Zhang G. - A Cross-Domain Recommender System With Kernel-Induced Knowledge Transfer for Overlapping Entities, in IEEE Transactions on Neural Networks and Learning Systems 30 (7) (2019) 1998-2012. doi:10.1109/ TNNLS.2018.2875144.
21. Badami M. and Nasraoui O. - Cross-Domain Hashtag Recommendation and Story Revelation in Social Media, 2018 IEEE International Conference on Big Data (Big Data), 2018, pp. 4294-4303, doi: 10.1109/BigData.2018.8622002.
22. Li Z., Qiao P., Zhang Y., and Bian K. - Adversarial Learning of Transitive Semantic Features for Cross-Domain Recommendation, 2019 IEEE Global Communications Conference (GLOBECOM), 2019, pp. 1-6, doi:10.1109/GLOBECOM38437. 2019.9013898.