IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 14, No. 3, June 2025, pp. 1839~1852
ISSN: 2252-8938, DOI: 10.11591/ijai.v14.i3.pp1839-1852

Heterogeneous semantic graph embedding assisted edge-sensitive learning for cross-domain recommendation

Pravin Ramchandra Patil, Pramod Halebidu Basavaraju
Department of Information Science and Engineering, Adichunchanagiri Institute of Technology affiliated with Visvesvaraya Technological University, Karnataka, India

Article Info

Article history:

Received Oct 7, 2024
Revised Feb 15, 2025
Accepted Mar 15, 2025

Keywords:

Attention method
Cross-domain recommendation
Edge pruning
Heterogeneous semantic graph embedding
Node2Vec
Word2Vec

ABSTRACT

In the digital age, recommendation systems navigate vast alternatives. Content-based, collaborative filtering, deep-driven, and cross-domain recommendation (CDR) have been studied significantly but face cold-start and data sparsity. Though CDR methods outperform others, they struggle to optimize user-item matrices. Recent graph-based CDR methods improve efficiency by leveraging additional user-item interactions; however, optimizing graph features remains an open research area. Moreover, current techniques do not consider the impact of noise items (unrelated) on recommendation accuracy. To address this gap, this paper develops a heterogeneous semantic graph-embedding (HSGE) edge-pruning model that leverages user ratings and item metadata in the source and target domains to recommend items to target domain users. To achieve it, at first Word2Vec method is applied to explicit and implicit details, followed by Node2Vec-driven graph embedding matrix generation. Our HSGE method obtains user-user, user-item, and item-item connections to achieve more semantic features. To improve accuracy, our model prunes edges that drop source domain items and allied edges unrelated to the target domain users. Subsequently, the retained HSGE matrices from both domains are processed for element-wise attention. A multi-layer perceptron with cosine similarity processed combined features matrices to generate top-N recommendations with superior hit-rate (HR) and normalized discounted cumulative gain (NDCG).

Corresponding Author:

Pravin Ramchandra Patil
Department of Information Science and Engineering
Adichunchanagiri Institute of Technology affiliated with Visvesvaraya Technological University
Belagavi 590018, Karnataka, India
Email: mr.patil1234@gmail.com

1. INTRODUCTION

Over the past few years, there has been high growth in digital data and allied applications serving various purposes, including business communication, and recommendation systems. As rising digital data volume requires robust analytics to help users navigate numerous options, as the challenge of choosing the right one can often lead to unexpected decisions [1], [2]. Over the rising population and allied digital data, recommender systems have gained widespread attention to alleviate challenges as mentioned earlier [3] in online digital ecosystems [4]. Recommender systems often exploit users' previous digital behavior, purchases, and preferences to recommend certain suitable items like products or services. Recommender systems are segmented into three primary categories [4]. These exist: i) collaborative filtering, ii) content-based recommendation, and iii) hybrid recommendation. The collaborative filter methods [5], [6] derive

user's preferences by exploiting their interaction with users and items without exploiting the user's or item's characteristics. On the contrary, content-based methods identify user preferences by assessing the similarity or shared features of the items that the user has brought or availed of [7]. Hybrid recommendation methods combine collaborative filtering and content-based recommenders, but achieving optimal solutions is challenging due to data sparsity issues [8], [9]. Presently, most at-hand recommender solutions rely on collaborative filtering methods [10] that hypothesize that users with similar interests or behaviors would have similar item choices. Though a history-based approach is easier to realize; yet, the allied data sparsity problem remains a bottleneck. Among the most employed item-based collaborative filtering methods [5] measure item similarity through item-user interactions. These methods rely on inner product (i.e., linear combination) between the user and item latent vector in a one-hot coding approach, which can limit performance and fail to capture complex interactions. Accuracy for new users is limited due to insufficient information about new entrants [7], a problem known as data sparsity. This issue can be addressed by amalgamating these methods, which employ both user reviews and item metadata for recommendation decisions.

Recently, different methods like topic modeling [11] and deep networks [12] have been designed to exploit content information to make recommendation decisions. Unlike these methods, cross-domain recommendation (CDR) methods [13] exploit user preferences and item characteristics from other domains to improve data in the receiving domain for recommendation. For instance, CDR can recommend movies based on a user's book purchases or suggest hotels based on travel history. The cross-domain recommender can exploit supplementary details from the varied domains to alleviate the problem of cold start and data sparsity [14]. A few approaches applied transfer learning methods [15], which were later applied to derive cross-domain collaborative filtering (CDCF) [16], [17] by transferring information from the originating domain to the receiving domain for an accurate recommendation based on matrix factorization (MF) [18], [19]. However, these methods primarily learn over the shallow and linear characteristics of the user and allied items. A few approaches apply deep learning methods for the latent feature extraction from users and items [1], [20]; yet, such methods employ one-hot vector as input and hence often fail in employing collaborative features between users and items. Such approaches are found suitable only for single-domain recommendation tasks [21]. Unlike single-domain recommendation models, CDRs are found more efficient in addressing data sparsity problems, where it can use various transfer learning methods [22], but they face challenges like costly, complex computation, and underutilization of in-domain data, affecting the recommender's accuracy.

New developments like CDR have recently addressed these problems by using insights from related areas, like recommending hotels based on travel history or movie recommendations based on book purchases. Deep learning approaches provide more complex feature extraction, whereas transfer learning and MF aid in information transfer across domains. Nevertheless, the efficacy and precision of these suggestions are still affected by obstacles such as computational complexity and the underutilization of domain data.

A literature review will examine previous work and key figures to highlight developments and issues in the field. The methodology section will describe the strategy and methods used, while the results section evaluates findings and potential improvements. Together, these sections clarify the present state and future of advanced recommendation systems.

2. LITERATURE SURVEY

The CDR methods are classified as semantic, factorial, tag, and graph-based [23]–[25]. However, among these approaches, the graph-based methods have shown better efficacy. In this method, users and items represent the nodes, and their inter-element (say, user-item) relationship is referred to as edges, where the inter-element relation and allied interactions can have a decisive impact on the recommendation accuracy. Some recent literature discusses the CDR systems exploiting reviews (text-data), transfer learning, deep learning, and graph-embedding approaches.

2.1. Text-data (review) based recommendation model

To address the data-sparsity problems, utilizing text data is crucial in recommendation systems. Srifi *et al.* [26] have used topic modeling to uncover hidden topics in user review text. A MF model called TopicMF was proposed, which combined user ratings and reviews to obtain latent topics. Social MF [27] and topic MF were employed to assess data fusion effectiveness for better recommendations. A probabilistic method [28] combined an integrated latent topic model with a random walk and restart technique for recommendation. These methods often ignore text order and use a bag-of-words (BOW) method to derive latent topic variables. Instead, embedding methods created word vectors, learned via convolutional neural networks (CNN) for prediction.

2.2. Deep-learning-based recommendation model

In the past, numerous efforts have been made by applying deep-learning methods for recommendation. Zhao [29] applied denoising auto-encoders to learn over the user-item matrix for user and item profiling. Xue *et al.* [30] developed an improved neural network-driven MF model, where at first a user-item matrix was generated over the explicit ratings and non-preference implicit review text. Once obtaining the user-item matrix, a deep neural network (DNN) was applied to learn the low-dimensional feature space (user-item matrix). Applying a binary loss function, their proposed deep network performed the recommendation. Research by Zhang *et al.* [31], a long short-term memory (LSTM)-based approach was applied to generate text features for further recommendation decisions. Research by Chen *et al.* [32], a neural attentional regression was applied in conjunction with review-level explanations for recommendation decisions, focusing on review text but ignoring item metadata, and primarily addressed single-domain recommendations.

2.3. Cross-domain recommendation

CDRs exploit the information from one domain to make recommendations in other domains. Cheema *et al.* [33] designed a latent user profiling approach that generates a user profile by gathering diverse user behaviors across multiple data domains. This domain-independent profile and contextual data enable relevant CDRs. The transfer learning method was applied to make serendipitous item recommendations [34]. A generalized cross domain-multi-dimension tensor factorization model to balance impact amongst domains effectively [35]. Hu *et al.* [36] applied a transfer meeting hybrid model for CDR by applying item reviews and article titles. They used attentively mined semantic features from review text to transfer vital information from an originating domain to perform learning and recommendation. Yet, these methods could not exploit user reviews and item metadata together. To alleviate the aforesaid limitation, the authors suggested transferring knowledge amongst the varied domains to make CDRs [3], [19]. An adaptive codebook transfer learning (ACTL) was proposed that identifies the suitable codebook scale to reduce computational cost and improve CDR [19]. Yu *et al.* [37] developed CDCF, for addressing data sparsity by utilizing rating information from two auxiliary domains: user-side and item-side, both characterized by dense rating data. The recommendation task is then reframed as a classification problem based on derived intrinsic features, which is solved using an SVM. The algorithm [38] addresses data sparsity in the receiving domain by augmenting user and item features with latent factors from both auxiliary domains, using Funk-singular value decomposition (SVD) to extract additional features. Two-sided CDCF algorithms [39], like two-stage stacked ensemble autoencoder (TSSEAE), enhance performance by combining user and item auxiliary domains, framing the CDCF problem as an ensemble of classifiers. The authors designed the framework in which one neural network was used to learn the reduced dimensional illustration from the one-hot coding of users and items, while another network leveraged auxiliary information in a different latent space [40].

Recently, a graph-based CDR method uses user-item interactions from the originating domain to recommend items in receiving. However, state-of-the-art graph-based recommenders may include irrelevant items from the originating domain, adding noise and reducing effectiveness. None of the existing methods could address the problem of noise items. To alleviate this, we propose a heterogeneous semantic graph-embedding (HSGE)-assisted model that prunes irrelevant edges in the originating domain. Our proposed method uses the Word2Vec semantic embedding method for the user's review, ratings, and item details. Then, the Node2Vec captures user-user, user-item, and item-item interactions from both domains. Here, the amalgamation of the different interactions constitutes the HSGE method. The model improves semantic feature embedding vectors by removing unrelated items from the originating domain for better learning and prediction. Using a multilayer feedforward neural network-multilayer perceptron (MFNN-MLP) with cosine similarity, it predicts top-N recommendations in the receiving domain, achieving better hit-rate (HR) and normalized discounted cumulative gain (NDCG) scores than other methods.

3. METHOD

The proposed method encompasses a heterogenous semantic graph embedding matrix, inter-domain relativeness-driven edge pruning, attention method, and MFNN-MLP for CDR. It processes user reviews and item details into input layers for each domain, creating semantic graph embeddings of user-user, user-item, and item-item interactions using Node2Vec. These embeddings are refined through edge pruning to remove unrelated items, followed by element-wise attention to form a combined feature model. This model is then used for learning and prediction via an MFNN-MLP with SoftMax for top-N recommendations at the output layer. The proposed model is depicted in Figure 1.

The cross-domain prediction model encompasses two domains, p (origin) and q (receiving), each with users and items containing explicit (user's rating and review) and implicit (item(s) details) details. By, the amalgamation of the review and the item's metadata creates a profile for a user u aimed at improving recommendation accuracy in the receiving domain. In real-time data models, there can be certain users that

overlap, which means certain users can exist in both the origin (say, p) and the receiving domain (say, q), serving as a bridge connecting both domains. This forms the basis for the cross-domain recommender design. The implementation schematic of the planned CDR model with five main components is shown in Figure 1. These are: i) input layer, ii) HSGE layer, iii) feature amalgamation layer, iv) MFNN-MLP layer, and v) prediction layer or the output layer.

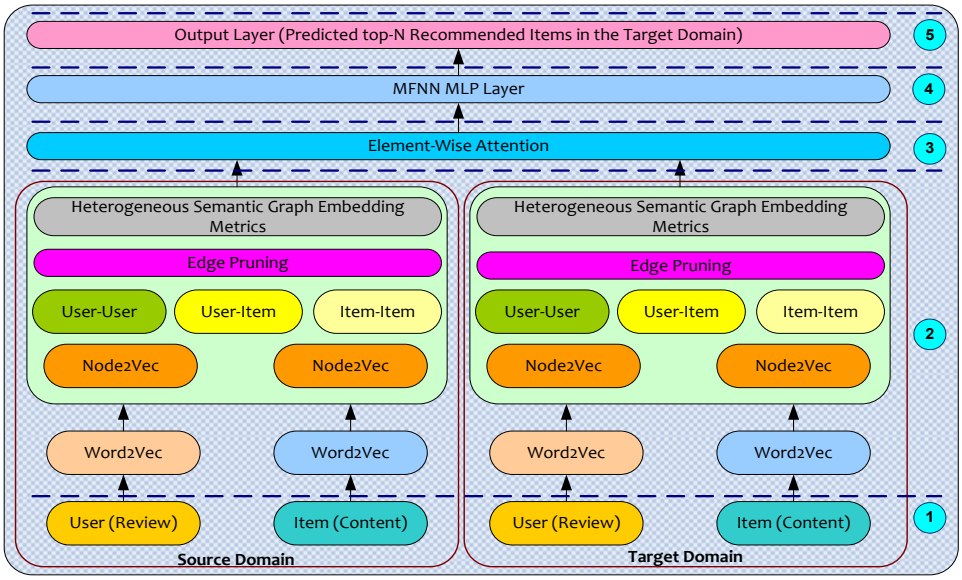

Figure 1. Proposed CDR model

3.1. Input layer

This work uses both explicit user details, like reviews and ratings, and implicit item information, such as metadata and feedback, as input. The explicit information constructed a profile for each user in both source as well as receiving domains. The user-item details were then fed as input to the semantic graph embedding layer.

3.2. Heterogenous semantic graph embedding layer

The user's review and item details from the originating (p) and the receiving domain (q) are fed as input to the semantic graph embedding layer. Unlike traditional one-hot encoding, which suffers from sparsity and noise issues, this novel graph-embedding method leverages user-user, item-user, and item-item interactions for improved predictions. It can be vital for CDR in both domains (i.e., p and q). Our proposed method uses heterogeneous graph embedding matrices for user-user, user-item, and item-item interactions in both source and receiving domains. Here, the user's rating and the item's content information related to each domain (i.e., p and q) are leveraged to constitute heterogenous graph embedding metrics. Unlike one-hot coding, Word2Vec offers richer semantic, and low-dimensional embedding metrics. We used Gensim's Word2Vec with skip-gram and negative sampling (SGNS) to transform content-review information of both p and q domains into corresponding semantic values, distinctly and also to transform each review or item metadata like titles, sub-titles, and descriptions into a semantic latent vector.

The Node2Vec method extracted HSGE metrics from latent embedding vectors by leveraging heterogeneous interactions, improving CDRs while addressing data sparsity and cold-start issues. The Node2Vec method extracted HSGE metrics from latent embedding vectors by leveraging heterogeneous interactions, improving CDRs while addressing data sparsity and cold-start issues. HSGE was obtained for both source and receiving domains, resulting in two heterogeneous graph embedding matrices. Noticeably, the HSGE graphs obtained contained nodes signifying users and items in the originating domain, while weighted edges represented the user's review and metadata content similarities obtained for the receiving domain, respectively. It improved the quality of user and item embedding matrices for learning and prediction. The homeland security and emergency management (HSEM) model consist of three functional elements: i) inter-domain relatedness-based edge pruning, ii) Doc2Vec (document) embedding, iii) semantic graph embedding and HSGE generation. The detail of these sub-components is given as follows.

3.2.1. Inter-domain relatedness-based edge pruning

This work introduces an inter-domain correlation-driven edge pruning model to improve recommendation accuracy by removing unrelated user-item edges in the receiving domain, preventing false positive results. To project aforesaid features to the latent embedding matrix or vector, the proposed model estimates projection vectors in both the originating domain as well as the receiving domain (their corresponding documents, i.e., d_s, d_t). Let, the projection matrices for originating and the receiving domain be $\Phi_s \in \mathbb{R}^{d_s \times d_{CCA}}$ and $\Phi_t \in \mathbb{R}^{d_t \times d_{CCA}}$. These projection vectors are obtained by performing canonical correlation analysis (CCA) over user's semantic embedding vector output (i.e., Word2Vec embedding vector for user) in both domains. In other words, let the user's embedding vector in originating and receiving domains be (1) and (2), respectively.

$$E_{u_s} = [e_s^1, e_s^2, \dots, e_s^U] \in \mathbb{R}^{d_s \times U} \tag{1}$$

$$E_{u_t} = [e_t^1, e_t^2, \dots, e_t^U] \in \mathbb{R}^{d_t \times U} \tag{2}$$

To achieve it, the proposed method performs maximization of the cost function defined in (3).

$$\left(\widehat{\Phi}_s, \widehat{\Phi}_t\right) = \arg\max_{\Phi_s, \Phi_t} \frac{\Phi_s^T C_{E_{Us}, E_{Ut}} \Phi_t}{\sqrt{\Phi_s^T C_{E_{Us}, E_{Us}} \Phi_s} \sqrt{\Phi_t^T C_{E_{Ut}, E_{Ut}} \Phi_t}} \tag{3}$$

In (3), the parameters Φ_s and Φ_t represents the projection embedding vectors, which can mathematically be derived as per (4) and (5), correspondingly.

$$\Phi_s = \mathbb{R}^{d_s \times d_{CCA}} \tag{4}$$

$$\Phi_t = \mathbb{R}^{d_t \times d_{CCA}} \tag{5}$$

The projection embedding metrics for source and receiving domain documents were obtained by maximizing the objective function defined in (3). The objective function (3) solved eigenvalue problem that eventually yielded (4) and (5) for the original and the receiving domains, respectively. The embedding matrix per domain was obtained as per the (6) and (7), conditioned with the (8) and (9), correspondingly.

$$\hat{e}_s^i \in \mathbb{R}^{d_{CCA}} \tag{6}$$

$$\hat{e}_t^i \in \mathbb{R}^{d_{CCA}} \tag{7}$$

$$\hat{e}_s^i = \Phi_s^T e_s^i \tag{8}$$

$$\hat{e}_t^i = \Phi_t^T e_t^i \tag{9}$$

The proposed model compared the items in originating and receiving domain in respective embedding latent space. Noticeably, only items with high correlation and similarity were retained. The noise proneness of the items in the originating domain applies the local density function over the semantic embedding metrics in source document d_s. In this manner, the difference between the local density of item i_s and the local density of the neighboring item o. The noise density for originating domain items is estimated as per (10).

$$LRD(i_s) = \frac{1}{\left(\frac{\sum_{o \in N_v(i_s)} Reach-dist_v(i_s,o)}{|N_v(i_s)|}\right)} \tag{10}$$

In (10), $N_v(i_s)$ represents a set of v adjoining items (in the vicinity of the source item i_s), while $LRD(i_s)$ be the inverse of the mean of the reachability distance via v nearest neighbors. Additionally, $Reach - dist_v(i_s, o)$ signifying the reachability distance between i_s and o is measured as per (11).

$$Reach - dist_v(i_s, o) = max\big(v - dist(i_s), d(i_s, o)\big) \tag{11}$$

In (11), $v - dist(i_s)$ represents the distance in between item i_s and the v-th neighboring items. Similarly, the distance between i_s and o items is given by $d(i_s, o)$. Assuming aforesaid outlier rank (10) as the (inter-item)

distance indicator (i.e., the distance between i_s and the item set, then the local reachability density of the item i_s is measured as per (12).

$$IOF(i_s) = \frac{\sum_{o \in N_v(i_s)} \frac{LRD(o)}{LRD(i_s)}}{|N_v(i_s)|} \quad (12)$$

In (12), the parameter $IOF(i_s)$ retrieves the extent to which the item i_s can be called as the noise or the outlier item having no significant relationship with the receiving domain user. Eventually, the semantic embedding feature for originating domain $\hat{e}_s^i \in \mathbb{R}^{d_{CCA}}$ is obtained after pruning the noisy or irrelevant edges in originating domain. Now, once obtaining the pruned item semantic vector in the originating domain, the graph embedding vectors or matrices were obtained for the user-user, user-item, and item-item interactions.

3.2.2. Doc2Vec embedding

The proposed model combines user-user, user-item, and item-item interaction relationships in heterogeneous semantic embedding matrices and is preserved as documents, creating unit graphs for both source and receiving domains. Content similarity between users and items is estimated using multi-source content information. The retained embedding metrics are converted into unit documents for each domain, collecting a semantic embedding review for each user. Being a multi-source document embedding method, the semantic embedding review C_{i*} for a user u_i is collected. For an $i-$th user (say, u_i), the allied user profile u_{pi} is obtained and saved in a single document d_i. For an item v_j, the reviews C_{*j} on the item and (item) detail id_j are collected in the same document d_{m+j}. Subsequently, the embedding vectors in aforesaid documents $D = \{d_1, d_2, \ldots, d_{m+n}\}$ are segmented by using $StanfordCoreNLP$ tool [41], which extracts user and item vectors from the input documents in both source and receiving domains. The Doc2Vec method was applied to map D into the text vectors UC and VC for users and items semantic embedded vectors.

3.3. Semantic graph construction

In this work, the users and items were mapped by exploiting their interaction relationships. The relative weight of each interaction was measured in sync with the local reachability density (12), which is derived as the normalized ratings (13).

$$Normalized\ rating\ for\ edges = \frac{R}{max(R)} \quad (13)$$

In reference to the targeted heterogenous (semantic) graph, the synthetic edges were obtained in between the two users or the two items based or corresponding normalized edge weights (13). In this manner, the likelihood $P(i,l)$ of the edge in between the two users $u_i$and u_j is estimated by using (14).

$$P(i,l) = \alpha.sim(UC_i, UC_l) \quad (14)$$

In (14), α refers to a hyper-parameter controlling the sampling likelihood, while $sim(UC_i, UC_l)$ represents the normalized cosine similarity (NCS) in between UC_i and UC_l. Be noted, here UC_i and UC_l represent the user profile documents for the $i-th$ and $l-th$ user. Similarly, the edges (probability) for item-item are also measured. Thus, based on the user-item, item-item and user-user interaction relationship, semantic heterogeneous graphs are obtained for both originating domain G^p and the receiving domain G^q.

3.4. Feature amalgamation layer

The semantic embedding matrix is estimated and combined using an element-wise attention mechanism. This creates an optimal weight set for common users learned across domains, with combined embeddings for the common users $\tilde{U}$ from each domain represents the features learned over the different fractions. In attention methods, a specific fraction of the feature representation is selected, which are subsequently given higher weights to generate the combined feature [42]. For a user u_i, it gives more attention to the elements possessing higher information from each pair of items in U_i^p and U_i^q. The proposed element-wise attention method generates two embedding matrices $\tilde{U}_i^p$ and $\tilde{U}_i^q$ for the common user u_i for the original and the receiving domain, correspondingly. The combined HSEM embedding feature $\tilde{U}_i^a$after attention mechanism for a user u_i in originating domain is derived using (15).

$$\tilde{U}_i^p = W^p \odot \tilde{U}_i^p + (1 - W^p) \odot U_i^q \quad (15)$$

In (15), $\odot$ represents the element-wise attention function, while $W^p \in \mathbb{R}^{m^p \times k}$ refers to the weight matrix retrieved at the attention layer for domain p. Thus, we obtained the cumulative embedding matrix for a user u_i as $\widetilde{U}_i^q$ for the receiving domain q. The combined embedding matrices from both domains are fed as input to the MLP layers (Figure 1) for learning and prediction. An MLP layer snippet is shown as follow.

3.5. Multi-layer perceptron layer

The study uses a fully-connected MLP to capture non-linear user-item interactions in each domain. This multi-layer feed-forward neural network learns complex interactions, unlike traditional methods that rely on linear element-wise attention. To learn over the latent features or HSEM embedding metrics of the users and items, the heterogeneous graph embedding matrix (features) from each domain is fed to the MLP that obtained the latent embedding matrix of the $p-th$ domain using (16).

$$\begin{aligned} X_0^p &= U_e^p \\ X_1^p &= f\left(W_1^X . X_0^p + B_1^X\right) \\ X_l^p &= f\left(W_l^X . X_{l-1}^p + B_l^X\right) \\ X^s &= f\left(W_{Lm}^X . X_{Lm-1}^p + B_{Lm}^X\right) \end{aligned} \tag{16}$$

In (16), $X_1^p, \ldots, X_{Lm}^p$ and $B_1^X, \ldots, B_{Lm}^X$ represent the weight and the bias matrices for the MLP. Here, Lm states the total number of MLP layers, while $f(*)$ be the rectified linear unit (ReLU) activation function. In (16), $X^p \in \mathbb{R}^{m \times c}$ be the user's latent feature matrix for the p-th domain learnt by the MLP method. Thus, the latent (semantic) embedding matrix for items pertaining to the $p-th$ domain is obtained as per (17).

$$\begin{aligned} Y_0^p &= V_e^p \\ Y_1^p &= f\left(W_1^Y . Y_0^p + B_1^p\right) \\ Y_l^p &= f\left(W_l^Y . Y_{l-1}^p + B_l^p\right) \\ Y^s &= f\left(W_{Lm}^Y . Y_{Lm-1}^p + B_{Lm}^p\right) \end{aligned} \tag{17}$$

In (17), $W_1^Y, \ldots, W_{Lm}^Y$ and $B_1^Y, \ldots, B_{Lm}^Y$ represent the weights and the bias matrices for the MLP network. $Y^s \in \mathbb{R}^{n_p \times c}$ be the item latent embedding metrics for the $p-th$ domain.

3.6. Output layer

This layer handles the user-item prediction for the receiving domains. The proposed model executes MLP training over the input combined user-item latent (say, HSGE) metrics, where the training model is designed based on the loss in between the predicted and the measured user-item interaction relationship. Thus, the proposed model is trained p by using the following objective function to make predictions for the receiving domain q. In (18), $l(y, \hat{y})$ refers a loss function between the observed user-item interactions (say, y) and the predicted user-item interaction $\hat{y}$. In (18), $Y^{p+}$and Y^{p-} represents the observed and the predicted user-item interactions over the originating domain p, correspondingly. $\|P^p\|_F^2 + \|Q\|_F^2$ refers the regularize, while λ represents the hyper-parameter controlling the level of significance of the regularize.

$$\min_{p^p, Q^p, \ominus^p} \sum_{y \in Y^{p+} \cup Y^{p-}} l(y, \hat{y}) + \lambda(\|P^p\|_F^2 + \|Q\|_F^2) \tag{18}$$

To alleviate over-fitting towards Y^+, a definite amount of the unobserved user-item interaction relationships is selected arbitrarily as the negative instances, given by $Y_{sampled}^-$. $Y_{sampled}^-$ was used to substitute Y^-, which helps achieving swift and accurate learning over non-linear input features. Thus, using user's review, for a user u_i and an item v_i, the corresponding user-item interactions y_{ij} was obtained (19).

$$y_{ij} = \begin{cases} r_{ij} & if\ y_{ij} \in Y^+ \\ 0 & if\ y_{ij} \in Y_{sampled}^- \\ null & Otherwise \end{cases} \tag{19}$$

In this work, a normalized cross-entropy loss function was applied to perform learning, which is derived as (20).

$$l(y, \hat{y}) = \frac{y}{max(R)} \log \hat{y} + \left(1 - \frac{y}{max(R)}\right) \log(1 - \hat{y}) \tag{20}$$

In (20), $max(R)$ represents the maximum rating on the originating domain a. In this work, MLP is used to represent a non-linear association amongst the users and items. Let, for the MLP network the input embedding matrices for the users and items over the originating domain be $M_{in}^{p} = [\tilde{U}^{p}; U^{pd}]$ and $Q_{in}^{p} = V^{p}$, correspondingly. Here, $\tilde{U}^{p}$ states the combined HSGE embedding matrix pertaining to the common users for the domain p, while the embedding matrix of the distinct users in domain p be U^{pd}. Thus, for user u_i and item v_j, the corresponding embedding matrices in the output layer of the MLP are obtained as (21) and (22).

$$M_{i}^{p} = M_{outi}^{p} = f\left(\ldots f\left(f\left(M_{in_i}^{p}.W_{M_1}^{p}\right).W_{M_2}^{p}\right)\right) \tag{21}$$

$$N_{j}^{p} = N_{outj}^{p} = f\left(\ldots f\left(f\left(N_{in_j}^{p}.W_{N_1}^{p}\right).W_{N_2}^{p}\right)\right) \tag{22}$$

In (21) and (22), ReLU activation function $f(*)$ was applied, while $W_{M_1}^{p}$ and $W_{M_2}^{p}$…, and $W_{M_1}^{p}$, $W_{N_2}^{p}$… be the weights of the multi-layer networks in varied layers in p domain for $M_{in_i}^{p}$ and $N_{in_j}^{p}$, correspondingly. Finally, in output layer, the predicted user-item interaction $\hat{y}_{i,j}$ between the user u_i and the item v_j on domain p is given by the (23).

$$\hat{y}_{i,j}^{p} = cosine\left(M_{i}^{p}, N_{j}^{p}\right) = \frac{M_{i}^{p}.N_{j}^{p}}{\|M_{i}^{p}\|\|N_{j}^{p}\|} \tag{23}$$

Similarly, the predicted interaction $\hat{y}_{i,j}^{p}$ on the receiving domain q were obtained to perform top-N recommendation. Thus, the cross-domain recommender model was developed using the aforementioned methods, as shown in Figure 2, with simulation results and insights discussed in the subsequent section.

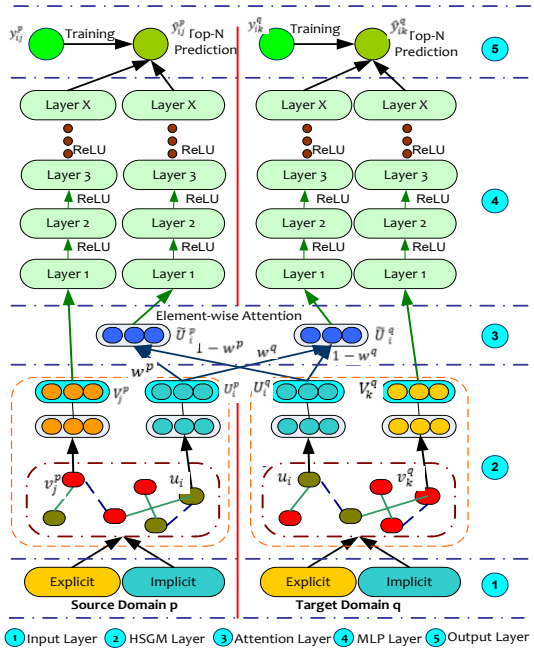

Figure 2. Proposed CDR model realization

4. RESULTS AND DISCUSSION

4.1. Experimental datasets

In alignment with the CDR task, this research considered two distinct benchmarks datasets. featuring real-time transaction details that varied significantly in data nature and item types. Specifically, the study focused on Douban datasets, which include book and movie subsets. We obtained the user and items with at least four interactions to exploit more behavioral aspects, reduce noise probability, and enhancing prediction accuracy in the originating domain by using the MFNN-MLP layer. A snippet of the dataset used

in this work is given in Table 1. As depicted in Table 1, the movie dataset encompassed a set of interactions with 2,718 users and 9,565 items that cumulatively gave rise to the data size of 1,133,420. Similarly, the book dataset possessed 2,718 users with 6,777 items with a total data size of 96,041. Our considered datasets encompassed both explicit data and implicit details, where the first embodies the reviews and ratings, while the latter contains item metadata. The proposed algorithm was developed in Python language, where the simulation environment encompassed Python 3.4, with the CPU armored with 8 GB memory running over Intel i5 processor. It also applied different Python libraries including NumPy, maths, SciPy, Gensim Word2Vec, and TensorFlow. The simulations were made on the Google Colab platform with the processor configured with the native graphical processing unit.

Table 1. Data specifications

Data	User	Items	Data size
Movie	2,718	9,565	1,133,420
Books	2,718	6,777	96,041

4.2. Simulation parameters

To ensure optimal performance, the proposed model employed numerous value additions in terms of the design parameters or the hyperparameter tuning. In the deployed recommender model, especially for HSEM, the sampling probability α was assigned as 0.05. In Gensim Word2Vec embedding of the ratings and items, we assigned window size as 3. The Node2Vec and Doc2Vec models' hyper-parameters were adjusted, and an MFNN-driven MLP was designed and tuned for different layer architectures. Initialized parameters were Gaussian distribution, and training was done with 8 negative instances per positive instance, 100 epochs, fixed learning rate, regularization parameter, and 64 batch sizes. Here, the key motive behind lower batch size was to improve (intrinsic) feature learning over the heterogenous embedding features which can have high non-linearity over millions of the user's profiles. Furthermore, we performed feature embedding concerning the different dimensions including 4, 8, 16, 32, and 64.

4.3. Evaluation parameters

The study focuses on a top-N recommendation task for users in a receiving domain, ranking target items based on interactions with uninteracted items. A leave-one-out approach is adopted, selecting recent interactions with a test item for each user. The performance parameters are HR and NDCG, which measure the recall rate and the quality of the ranking. The aim is to evaluate the effectiveness of this ranking-based method. The proposed cross-domain recommender model generates a (top-N) ranked list of items (in the receiving domain) for the target user. If a target item is in this list, it's a hit. We define hit rate (HR) as in (24). In (24), the denominator $|T|$ represents the total number of interactions in the test set. In this work, NDCG was applied to provide (or assess) the hit position by assigning higher scores to the hits (especially at the top K ranks).

$$HR@K = \frac{No.of\ Hits@K}{|T|} \tag{24}$$

Mathematically, we derived NDCG@K as per the (25). In (25), the parameter r_i states the ranked relevance of the target item at the i-th position. Therefore, in this context, $r_i = 1$.

$$NDCG@K = \sum_{i=1}^{K} \frac{2^{r_i}-1}{\log_2(i+1)} \tag{25}$$

In otherwise case, we label $r_i = 0$. Additionally, we also estimated the root mean square error (RMSE) performance, which was derived as per the (26). In (26), T represents the total number of test ratings, while $R^s(i,j)$ states the real rating and the measured or predicted rating be $\hat{R}^s(i,j)$.

$$RMSE = \sqrt{\frac{1}{T}\sum_{s,i,j}\left(R^s(i,j) - \hat{R}^s(i,j)\right)^2} \tag{26}$$

This work characterizes overall performance and involves both intra- and inter-model assessments. Here, the intra-model assessment states the performance characterization with the different embedding dimensional (say, latent dimensions). These assessments are conducted across different input datasets to analyze performance trends. These assessments are conducted across different input datasets.

4.3.1. Intra-model assessment

To evaluate the effectiveness of the inter-domain relatedness-driven edge pruning method, we simulated our model with and without edge pruning (by feeding direct HSGE metrics to the attention layer). In addition, we evaluated various latent dimensions (4, 8, 16, 32, and 64), with simulation results for the Douban book and movie datasets presented in Table 2. It is often hypothesized that the high dimensional features often yield superior results but increased computation. we simulated our CDR model using five embedding dimensions (k=4, 8, 16, 32, 64), where k is the embedding dimensions. The simulation results in Table 2 show that as k increases, the HR@K also rises. Training on various input data subsets, including Book and Movie, shows that increasing latent and intrinsic feature learning improves both HR and NDCG. As depicted in Table 2, the average HR@4 was 0.5675, while NDCG@4 was observed as 0.3470. Similarly, in another test case with HSGE embedding dimension of 8, we measured average HR@8 as 0.5791, while NDCG@8 was found as 0.3805. For k=16, we found HR@16 as 0.5799 and NDCG@16 as 0.3990. On the contrary, for $k = 32$, the model yielded HR@32 as 0.5998 and NDCG@32 as 0.4171. Similarly, over simulation, it depicted HR@64 of 0.6070 and NDCG@64 as 0.3796. Table 3 shows the average HR and NDCG across various embedding dimensions, with higher sizes leading to decreased RMSE, demonstrating the model's robustness and efficiency in providing accurate top-N recommendations. The results indicate that higher embedding dimensions lead to better HR and NDCG for reliable CDRs. A consistent efficiency pattern across datasets (books and movies) confirms the model's suitability for real-time recommendation tasks.

To evaluate the impact of inter-domain edge pruning on top-N recommendations, we simulated our model with and without pruning. Results are shown in Table 4. The result in Table 4 can easily be inferred that the use of the proposed edge pruning method with HSGE mechanism enables superior HR and corresponding NDCG performance. The average HR performance increased from 0.5410 without edge-pruning to 0.5866 with edge-pruning. It reveals that the use of our proposed edge pruning model with HSGE enables almost 9.2% better performance (i.e., HR). The results confirm that the edge pruning method with the HSGE model outperforms methods relying solely on user-item interactions, as it reduces noise, significantly improving both HR and RMSE shown in Tables 3 and 4. The HSGE-CDR model's performance was evaluated against other leading methods. The results are detailed in the subsequent section.

Table 2. Performance over the different latent embedding dimensions

Latent dimensions or embedding size (K)	Data	HR@K	NDCG@K	RMSE
4	Book	0.4531	0.2831	1.409
	Movie	0.6820	0.4109	1.389
8	Book	0.4861	0.2994	1.091
	Movie	0.6721	0.4617	1.077
16	Book	0.4824	0.2993	0.987
	Movie	0.6775	0.4987	0.988
32	Book	0.5015	0.3740	0.938
	Movie	0.6982	0.4603	0.874
64	Book	0.5126	0.3779	0.880
	Movie	0.7014	0.4813	0.871

Table 3. Average HR and NDCG performance over the different latent dimensions

Latent dimensions or embedding size (K)	Average		
	HR	NDCG	RMSE
4	0.5675	0.3470	1.399
8	0.5791	0.3805	1.084
16	0.5799	0.3990	0.987
32	0.5998	0.4171	0.906
64	0.6070	0.4296	0.875

Table 4. Efficacy of edge pruning on HSGE driven CDR

Latent dimensions or embedding size (K)	Data	Without edge pruning			With edge-pruning		
		HR@K	NDCG@K	RMSE	HR@K	NDCG@K	RMSE
4	Book	0.3831	0.2189	1.998	0.4531	0.2831	1.409
	Movie	0.5993	0.3776	1.849	0.6820	0.4109	1.389
8	Book	0.4473	0.2851	1.248	0.4861	0.2994	1.091
	Movie	0.6110	0.4294	1.216	0.6721	0.4617	1.077
16	Book	0.4361	0.2840	1.198	0.4824	0.2993	0.987
	Movie	0.6485	0.4401	1.233	0.6775	0.4987	0.988
32	Book	0.4791	0.3392	1.194	0.5015	0.3740	0.938
	Movie	0.6620	0.4261	1.130	0.6982	0.4603	0.874
64	Book	0.4899	0.3482	1.078	0.5126	0.3779	0.880
	Movie	0.6538	0.3399	1.004	0.7014	0.4813	0.871

4.3.2. Inter-model assessment

To assess the performance of the proposed CDR approach, we analyzed results from various state-of-the-art methods at embedding sizes of 8, 16, 32, and 64. The considered reference methods employ cosine similarity to perform top-N recommendations in cross-domain applications. Typically, the cosine similarity method the final component in interaction association analysis for a recommendation, where it provides K estimates of most similar items, whereas the cosine similarity method is applied to measure similarity amongst the items. We assessed performance using HR and NDCG metrics. Before discussing the empirical results and their implications, we briefly outline these methods:

- GMF [20]: it represents a generalized form of the MF method, utilizing a sophisticated neural network model that learns user-item interaction relationships by employing certain activation functions over the linear combination of the element-wise multiplication of the input vectors. This approach receives user identification and corresponding items as one-hot vectors (that often impose computational exhaustion). According to Ma *et al.* [2], the retrieved one-hot vectors of the users and items are projected as input to the dense vectors, which are used as input to the neural network to obtain the prediction scores for the user-item pairs.
- Deep matrix factorization (DMF) [30]: the method generates an implicit user-item embedding matrix, which is input to a DNN for high-dimensional user and item vectors. The vectors are projected to a low-dimensional latent space, and the cosine similarity method is applied for recommendation results. Xue *et al.* [30] employed a dual-layer DMF method to generate recommendation results.
- $NeuMF_C$: it is an extended approach of $NeuMF$ model that employs clustering as a component to make recommendation results.
- $NeuMF_{mp}$: this approach applies Word2Vec on user-item details to generate a semantic vector, which is then applied as input to the neural network to perform recommendation results. Unlike $NeuMF$, this method applies Word2Vec embedded feature vector as input to the $NeuMF$ to perform recommendation.

Table 5 presents the HR@K performance over the different embedding dimensions, indicating that larger embedding sizes improve HR performance, with HR@64 showing superior over the other embedding sizes. The GMF-based CDR model performs the maximum HR for HR@64 as 0.2901 and 2997 for book and movies data subsets, respectively. On the other hand, the DMF-based model exhibits maximum performance for HR@64 and was found as 0.3004 and 0.2899 with Book and Movie datasets, respectively. The $NeuMF_C$ method too exhibits superior with K=64, where it achieves 0.2832 and 0.3421, correspondingly with book and movie datasets. $NeuMF_{mp}$ as an improved solution than the $NeuMF_C$ method. The simulation reveals that the $NeuMF_{mp}$ method exhibits HR@64 of 0.5126 and 0.7014 for book and movie datasets, respectively. In comparison to the state-of-art methods (i.e., GMF, DMF, $NeuMF_C$, and $NeuMF_{mp}$), our proposed method shows superior results of HR@64 as 0.3779 and 0.4813 for book and movie. Similar performances were obtained with other embedding sizes as well. The NDCG performance in Table 6 reveals a similar performance as that of HT@K. As depicted in the following result in Table 6, the NDCG@K performance reveals that the proposed CDR model performance is superior over other approaches (NDCG@64 of 0.3779 and 0.4813, respectively over book and movie datasets). Interestingly, the depth assessment also reveals that, unlike K=64, our proposed method exhibits superior outputs even with lower K (i.e., K=4, 8, 16). The results in Table 5 indicate that the average HR@K for K=4, 8, and 16 exhibits 0.5675, 0.5791 and 0.5799, respectively. Considering K=8 as a moderate embedding size (assumption), our depth analysis reveals that the GMF and DMF methods show.

Table 5. Relative HR@K efficacy analysis

Method	Data	HR@K				
		K=4	K=8	K=16	K=32	K=64
GMF	Book	0.0985	0.1094	0.1375	0.2017	0.2901
	Movie	0.1246	0.1898	0.2311	0.2458	0.2997
DMF	Book	0.1304	0.1969	0.2119	0.2499	0.3004
	Movie	0.1332	0.2331	0.2398	0.2503	0.2899
$NeuMF_C$	Book	0.1731	0.2316	0.2390	0.2570	0.2832
	Movie	0.1996	0.2402	0.2441	0.2898	0.3421
$NeuMF_{mp}$	Book	0.2742	0.4362	0.2973	0.3117	0.3568
	Movie	0.2995	0.4962	0.3991	0.3996	0.5310
Proposed Method	Book	0.4531	0.4861	0.4824	0.5015	0.5126
	Movie	0.6820	0.6721	0.6775	0.6982	0.7014

Average NDCG@8 as 0.2776, and 0.2604, respectively. While $NeuMF_C$ and $NeuMF_{mp}$ methods output NDCG@8 as 0.2947 and 0.301, respectively. In comparison to these results, our proposed model

exhibits an average NDCG of 0.3805, which is higher than any other existing approaches and hence confirms the superiority of the suggested solution over other state-of-the-art approaches About the HR@8 as well, GMF, DMF, $NeuMF_C$, and $NeuMF_{mp}$ show outputs as 0.1496, 0.215, 0.2359, and 0.4662. On the contrary, our proposed CDR model exhibits an average HT@8 of 0.5791, which outperforms any other method. Thus, it confirms the superiority of the proposed model over other state-of-the-art models.

Table 6. Relative NDCG efficacy analysis

Method	Data	NDCG@K				
		K=4	K=8	K=16	K=32	K=64
GMF	Book	0.2300	0.2413	0.3318	0.3384	0.3460
	Movie	0.3167	0.3139	0.3660	0.3341	0.3952
DMF	Book	0.2416	0.2508	0.2440	0.3312	0.3229
	Movie	0.2667	0.2701	0.2510	0.2351	0.3550
$NeuMF_C$	Book	0.2412	0.2811	0.2615	0.3691	0.3212
	Movie	0.3046	0.3084	0.2801	0.3416	0.3666
$NeuMF_{mp}$	Book	0.2615	0.2893	0.2729	0.3711	0.3991
	Movie	0.3247	0.3127	0.2842	0.3415	0.3779
Proposed method	Book	0.2831	0.2994	0.2993	0.3740	0.3779
	Movie	0.4109	0.4617	0.4987	0.4603	0.4813

5. CONCLUSION

This paper presents a novel HSGE-assisted edge-sensitive learning approach for cross-domain recommender design. The model uses user ratings and item metadata in both original and receiver domains to recommend items for target users. The model employs the Word2Vec method for explicit and implicit details, followed by Node2Vec-driven graph embedding matrix generation. The HSGE method obtains user-user, user-item, and item-item interactions for additional representative semantic features. The model also employs an inter-domain relatedness-sensitive edge pruning method to improve accuracy. The model's superior HR and NDCG results make it ideal for real-time cross-domain recommender tasks without cold-start and sparsity problems. The model can alleviate cold-start and sparsity problems and maintain high CDR efficiency with low computational cost or complexity. Further, the enhanced CDR can be optimized through advancements in transfer learning techniques, emphasizing the development of more sophisticated models that effectively capture inter-domain relationships. Exploring domain adaptation techniques and employing graph neural networks to model complex cross-domain interactions may improve recommendation accuracy.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration. According to the CRediT, the following table presents the individual author contributions.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Pravin Ramchandra Patil	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓		✓	
Pramod Halebidu Basavaraju		✓				✓	✓	✓	✓	✓	✓	✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability does not apply to this paper, as no new data were created. The dataset used in this study is publicly available at Kaggle.

REFERENCES

[1] C. D. Wang, Z. H. Deng, J. H. Lai, and P. S. Yu, "Serendipitous recommendation in e-commerce using innovator-based collaborative filtering," *IEEE Transactions on Cybernetics*, vol. 49, no. 7, pp. 2678–2692, 2019, doi: 10.1109/TCYB.2018.2841924.

[2] J. Ma, J. Wen, M. Zhong, W. Chen, and X. Li, "MMM: multi-source multi-net micro-video recommendation with clustered hidden item representation learning," *Data Science and Engineering*, vol. 4, no. 3, pp. 240–253, 2019, doi: 10.1007/s41019-019-00101-4.

[3] L. Huang, Z. L. Zhao, C. D. Wang, D. Huang, and H. Y. Chao, "LSCD: low-rank and sparse cross-domain recommendation," *Neurocomputing*, vol. 366, pp. 86–96, 2019, doi: 10.1016/j.neucom.2019.07.091.

[4] S. Pyo, E. Kim, and M. Kim, "LDA-based unified topic modeling for similar TV user grouping and TV program recommendation," *IEEE Transactions on Cybernetics*, vol. 45, no. 8, pp. 1476–1490, 2015, doi: 10.1109/TCYB.2014.2353577.

[5] V. Kumar, A. K. Pujari, S. K. Sahu, V. R. Kagita, and V. Padmanabhan, "Collaborative filtering using multiple binary maximum margin matrix factorizations," *Information Sciences*, vol. 380, pp. 1–11, 2017, doi: 10.1016/j.ins.2016.11.003.

[6] C. Shi *et al.*, "Deep collaborative filtering with multi-aspect information in heterogeneous networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 4, pp. 1413–1425, 2021, doi: 10.1109/TKDE.2019.2941938.

[7] D. Wang, Y. Liang, D. Xu, X. Feng, and R. Guan, "A content-based recommender system for computer science publications," *Knowledge-Based Systems*, vol. 157, pp. 1–9, 2018, doi: 10.1016/j.knosys.2018.05.001.

[8] A. K. Sahu and P. Dwivedi, "User profile as a bridge in cross-domain recommender systems for sparsity reduction," *Applied Intelligence*, vol. 49, no. 7, pp. 2461–2481, 2019, doi: 10.1007/s10489-018-01402-3.

[9] L. Hu, S. Jian, L. Cao, Z. Gu, Q. Chen, and A. Amirbekyan, "HERS: modeling influential contexts with heterogeneous relations for sparse and cold-start recommendation," *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, pp. 3830–3837, 2019, doi: 10.1609/aaai.v33i01.33013830.

[10] Y. Zhang, C. Yin, Q. Wu, Q. He, and H. Zhu, "Location-aware deep collaborative filtering for service recommendation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 6, pp. 3796–3807, 2021, doi: 10.1109/TSMC.2019.2931723.

[11] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: a factorization-machine based neural network for CTR prediction," *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pp. 1725–1731, 2017, doi: 10.24963/ijcai.2017/239.

[12] O. Barkan, A. Caciularu, O. Katz, and N. Koenigstein, "Attentive Item2vec: neural attentive user representations," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 3377-3381, doi: 10.1109/ICASSP40776.2020.9053071.

[13] H. Zhao, Q. Yao, J. Li, Y. Song, and D. L. Lee, "Meta-graph based recommendation fusion over heterogeneous information networks," *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 635–644, 2017, doi: 10.1145/3097983.3098063.

[14] L. Hu, L. Cao, J. Cao, Z. Gu, G. Xu, and J. Wang, "Improving the quality of recommendations for users and items in the tail of distribution," *ACM Transactions on Information Systems*, vol. 35, no. 3, 2017, doi: 10.1145/3052769.

[15] L. Nie, X. Song, and T.-S. Chua, *Learning from multiple social networks*. Cham: Springer International Publishing, 2016, doi: 10.1007/978-3-031-02300-2.

[16] H. Kanagawa, H. Kobayashi, N. Shimizu, Y. Tagami, and T. Suzuki, "Cross-domain recommendation via deep domain adaptation," *Advances in Information Retrieval*, pp. 20–29, 2019, doi: 10.1007/978-3-030-15719-7_3.

[17] A. K. Sahu and P. Dwivedi, "Knowledge transfer by domain-independent user latent factor for cross-domain recommender systems," *Future Generation Computer Systems*, vol. 108, pp. 320–333, 2020, doi: 10.1016/j.future.2020.02.024.

[18] A. K. Sahu and P. Dwivedi, "Matrix factorization in cross-domain recommendations framework by shared users latent factors," *Procedia Computer Science*, vol. 143, pp. 387–394, 2018, doi: 10.1016/j.procs.2018.10.410.

[19] M. He, J. Zhang, and S. Zhang, "ACTL: adaptive codebook transfer learning for cross-domain recommendation," *IEEE Access*, vol. 7, pp. 19539–19549, 2019, doi: 10.1109/ACCESS.2019.2896881.

[20] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, "Neural collaborative filtering," *26th International World Wide Web Conference,* pp. 173–182, 2017, doi: 10.1145/3038912.3052569.

[21] Z. H. Deng, L. Huang, C. D. Wang, J. H. Lai, and P. S. Yu, "DeepCF: a unified framework of representation learning and matching function learning in recommender system," *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, pp. 61–68, 2019, doi: 10.1609/aaai.v33i01.330161.

[22] W. Pan, "A survey of transfer learning for collaborative recommendation with auxiliary data," *Neurocomputing*, vol. 177, pp. 447–453, 2016, doi: 10.1016/j.neucom.2015.11.059.

[23] C. Zhao, C. Li, and C. Fu, "Cross-domain recommendation via preference propagation graphnet," in *CIKM '19: Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 2165–2168, doi: 10.1145/3357384.3358166.

[24] C. Gao *et al.*, "Cross-domain recommendation without sharing user-relevant data," in *WWW '19: The World Wide Web Conference*, 2019, pp. 491–502, doi: 10.1145/3308558.3313538.

[25] D. Yang, J. He, H. Qin, Y. Xiao, and W. Wang, "A graph-based recommendation across heterogeneous domains," in *CIKM '15: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015, pp. 463–472, doi: 10.1145/2806416.2806523.

[26] M. Srifi, A. Oussous, A. A. Lahcen, and S. Mouline, "Recommender systems based on collaborative filtering using review texts-A survey," *Information*, vol. 11, no. 6, 2020, doi: 10.3390/INFO11060317.

[27] G. N. Hu, X. Y. Dai, Y. Song, S. J. Huang, and J. J. Chen, "A synthetic approach for recommendation: combining ratings, social relations, and reviews," *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pp. 1756–1762, 2015.

[28] S. Feng, J. Cao, J. Wang, and S. Qian, "Recommendations based on comprehensively exploiting the latent factors hidden in items' ratings and content," *ACM Transactions on Knowledge Discovery from Data*, vol. 11, no. 3, 2017, doi: 10.1145/3003728.

[29] J. Zhao, L. Wang, D. Xiang, and B. Johanson, "Collaborative denoising auto-encoders for top-N recommender systems," SIGIR | Special Interest Group on Information Retrieval, Paris, France. Amazon Science, pp. 1-6, 2016.

[30] H. J. Xue, X. Y. Dai, J. Zhang, S. Huang, and J. Chen, "Deep matrix factorization models for recommender systems," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*, pp. 3203–3209, 2017, doi: 10.24963/ijcai.2017/447.
[31] Q. Zhang, J. Wang, H. Huang, X. Huang, and Y. Gong, "Hashtag recommendation for multimodal microblog using co-attention network," *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*, pp. 3420–3426, 2017, doi: 10.24963/ijcai.2017/478.
[32] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Neural attentional rating regression with review-level explanations," in *WWW '18: Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 1583–1592, doi: 10.1145/3178876.3186070.
[33] A. A. Cheema, M. S. Sarfraz, M. Usman, Q. Uz Zaman, U. Habib, and E. Boonchieng, "KT-CDULF: knowledge transfer in context-aware cross-domain recommender systems via latent user profiling," *IEEE Access*, vol. 12, pp. 102111–102125, 2024, doi: 10.1109/ACCESS.2024.3430193.
[34] G. Pandey, D. Kotkov, and A. Semenov, "Recommending serendipitous items using transfer learning," in *International Conference on Information and Knowledge Management, Proceedings*, 2018, pp. 1771–1774, doi: 10.1145/3269206.3269268.
[35] A. Taneja and A. Arora, "Cross domain recommendation using multidimensional tensor factorization," *Expert Systems with Applications*, vol. 92, pp. 304–316, 2018, doi: 10.1016/j.eswa.2017.09.042.
[36] G. Hu, Y. Zhang, and Q. Yang, "Transfer meets hybrid: a synthetic approach for cross-domain collaborative filtering with text," in *WWW '19: The World Wide Web Conference*, 2019, pp. 2822–2829, doi: 10.1145/3308558.3313543.
[37] X. Yu, Y. Chu, F. Jiang, Y. Guo, and D. Gong, "SVMs classification based two-side cross domain collaborative filtering by inferring intrinsic user and item features," *Knowledge-Based Systems*, vol. 141, pp. 80–91, 2018, doi: 10.1016/j.knosys.2017.11.010.
[38] X. Yu, F. Jiang, J. Du, and D. Gong, "A cross-domain collaborative filtering algorithm with expanding user and item features via the latent factor space of auxiliary domains," *Pattern Recognition*, vol. 94, pp. 96–109, 2019, doi: 10.1016/j.patcog.2019.05.030.
[39] X. Yu, Q. Peng, L. Xu, F. Jiang, J. Du, and D. Gong, "A selective ensemble learning based two-sided cross-domain collaborative filtering algorithm," *Information Processing and Management*, vol. 58, no. 6, 2021, doi: 10.1016/j.ipm.2021.102691.
[40] J. He, R. Liu, F. Zhuang, F. Lin, C. Niu, and Q. He, "A general cross-domain recommendation framework via bayesian neural network," in *2018 IEEE International Conference on Data Mining (ICDM),* Singapore, 2018, pp. 1001–1006, doi: 10.1109/ICDM.2018.00125.
[41] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The stanford CoreNLP natural language processing toolkit," in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Baltimore, Maryland, 2014, pp. 55–60, doi: 10.3115/v1/p14-5010.
[42] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, CoRR, 2015.

BIOGRAPHIES OF AUTHORS

Pravin Ramchandra Patil received a degree in Bachelor of Information Science and Engineering from Shivaji University, Kolhapur, Maharashtra, India, and M.E. in computer science and engineering from Shivaji University, Kolhapur, Maharashtra, India. He is pursuing a Ph.D. in information science and engineering at Adichunchanagiri Institute of Technology, Chikkamagaluru affiliated with Visvesvaraya Technological University, Belgaum Karnataka, India. He is currently working as an assistant professor in the Department of Information Technology at Vasantdada Patil College of Engineering and Visual Arts, Sion, Mumbai, Maharashtra, India. He has 14 years of teaching experience. His research interest includes recommendation system, artificial intelligence, deep learning, and machine learning. She has published papers in conferences and international journals. He can be contacted at email: mr.patil1234@gmail.com or pravin_patil@pvppcoe.ac.in.

Dr. Pramod Halebidu Basavaraju has an experience of 12 and above years as an academician, currently working as an associate professor in the Department of Information Science and Engineering, Adichunchanagiri Institute of Technology, Chikkamagaluru affiliated with Visvesvaraya Technological University, Karnataka, India. In his credit there are 19 research papers were published in reputed journals, 9 research papers, and 10 papers have been presented at international and national conferences respectively. He received a Bachelor of Engineering degree in computer science and engineering from the Visvesvaraya Technological University in 2007, and a Master of Technology degree in computer science from the University of Mysore in 2012. He received a doctorate, Ph.D. in the field of wireless sensor networks from the Department of Computer Science and Engineering, Shri Venkateshwara University, Uttar Pradesh in 2019. His research area includes wireless sensor networks, network security, and data analytics. He can be contacted at email: hbpramod@aitckm.in.