



## Stratified Advance Personalized Recommendation System Based on Deep Learning

Arpit Deo<sup>1</sup>, Riya Jaisinghani<sup>1\*</sup>, Sagar Gupta<sup>1</sup>, Safdar Sardar Khan<sup>2</sup>, Adish Soni<sup>1</sup>, Kushal Gehlot<sup>3</sup>

<sup>1</sup> Department of Computer Science and Engineering, Medi-Caps University, Indore 453331, India

<sup>2</sup> Department of Computer Science and Engineering, RKDF Institute of Science and Technology, Bhopal 462033, India

<sup>3</sup> Department of Management, Astral Institute of Technology and Research, Indore 452020, India

Corresponding Author Email: [en20cs304048@medicaps.ac.in](mailto:en20cs304048@medicaps.ac.in)

<https://doi.org/10.18280/isi.280121>

### ABSTRACT

**Received:** 13 November 2022

**Accepted:** 30 January 2023

#### Keywords:

*E-Commerce recommendation system, deep learning, transfer learning, cross domain analysis, user behavior analysis, user feedback analysis, re-ranking*

A recommendation system is a refinement system that uses massive amounts of data to forecast and present user-preferred products. We employ web log files, including previously searched data, and browsing history, and transmit it to a SoftMax model in our recommendation model. We also use this data to perform user behaviour analysis using the k means clustering algorithm. Furthermore, we transfer users' feedback data, which is divided into explicit and implicit data, to the EIMNF model, which is a neural matrix model that aids us in forecasting users' preferences. Furthermore, we undertake cross domain analysis with the use of a CNN FT model, and all of the outputs created by the algorithm are referred to as intermediate recommended items, and they are delivered to an item pool to be reranked. Re-ranking improves accuracy and allows us to provide the best possible suggestions to our users. We employ a graph neural network to execute re-ranking, and the best things generated after reranking are provided to our end-user. We compared our model to various models and found that proposed model has 0.91 precision, 0.84 recall, 0.87 F-Measure and holds 91% accuracy.

## 1. INTRODUCTION

Recommendation systems and e-commerce go hand in hand. It's good to know that the majority of recommendations are to our taste in this new digital world when everything is now available online. E-commerce appears to benefit the most from recommendation algorithms. These technologies, which are based on artificial intelligence and deep learning, are critical in today's digital market. Because we don't have to go through a lot of browsing pages, this technique makes searching a lot easier. Appropriate suggestions lead to larger carts in online stores like Amazon, Flipkart, and others, significantly increasing e-business margins and generating revenue for online platforms.

The artificial intelligence strategy used in recommendation systems, which has considerably improved performance, includes fuzzy technique, transfer learning, genetic algorithms, evolutionary algorithms, neural networks and deep learning, and active learning [1]. The origins of the recommendation system can be traced back to LIRA [2], which pioneered tailored recommendations. Initially, the recommendation system relied on collaborative and content-based filtering approaches. However, in today's digital age, with the rapid growth of "BIG DATA," traditional strategies have proven ineffective, prompting the development of a hybrid recommendation system combining the two to provide recommended items based on the user's similar neighbours [3]. However, there were still other hurdles. Every day, a large number of new users and goods are uploaded, resulting in cold start issues [4] because the system lacks sufficient data or past rating data to know which items to promote. Overspecialization is another issue that arises [5]. The

recommender system favors items that the user is already familiar with, obstructing new and perhaps better options.

Deep learning, a machine learning technique, has now taken over major traditional methods of a recommender model. Deep learning techniques have proved to be more efficient as a non-linear model as compared to that of linear models of a recommendation system [6]. It uses the stochastic gradient descent (SGD) algorithm of any variant to optimize the outcomes. there are three main techniques used in recommender models. Firstly, Auto-encoder - has a better understanding of the user demands and item features, thus leading to higher recommendation accuracy than traditional models [7]. It is an unsupervised learning technique that uses back propagation setting the output target values to be equal to the input values. An encoder is built from the input layer and the hidden layer. A decoder is built using hidden layer and the output layer. Between these two layers, lies a code which represents the compressed input provided to the decoder [8]. Emre Tercan's study has demonstrated the efficiency of autoencoders for extracting lake areas from high-resolution RGB image. These experiments show that autoencoders perform similarly to conventional image segmentation methods when it comes to reliably identifying lake regions in high-resolution data [9]. The second, being convolutional neural networks (CNN). They integrate feature extraction function into multilayer perceptron by restructuring structure and reducing weight and omit the complex process of image feature extraction prior to recognition [10]. Convolutional neural networks are now included into the embedded feature selection technique and the new Fuzzy Temporal Logic based Decision Tree classifier [11] Thirdly, a recurrent neural network (RNN), it can learn highly complex relationships

from a given sequential data. RNNs can be trained to predict the next item a user is likely to interact with based on their past behavior. For example, an RNN-based recommendation system may consider the items a user has recently viewed, the items they have added to their shopping cart, and the items they have purchased in the past. The RNN can then generate a personalized recommendation for the user based on their past behavior patterns [12].

We developed an artificial intelligence-based recommendation mechanism. The advent of a deep neural network is a watershed moment in the history of recommendation systems. We present a hybrid artificial intelligence based recommendation system that utilizes both machine learning and deep learning approaches. We produce recommended items by processing users' history and weblog files using a SoftMax training model. Following that, clustering is used to examine the user's activity [13]. Additionally, we mine user preferences using the transfer learning approach and cross-domain analysis using CNN\_FT [14]. This also resolves the long tail issues that frequently plague recommendation systems. Additionally, we analyse user input using the EIMNF model [15]. Following that, we do re-ranking on the above-mentioned intermediate rated items using a graph neural network [16].

## 2. RELATED WORK

The recommendation system is a critical component of current e-commerce platforms such as Amazon, Flipkart, and Netflix. Predicting user-satisfied recommended items is very algorithm- and model dependent. Traditional recommendation systems were widely categorized into content-based, collaborative filtering, and hybrid recommendation systems [17]. Content-based recommender systems generate a user profile containing the user's preferences, which is then supplied to machine learning algorithms [18]. This profile is then matched to the qualities of the item and a suggestion list is generated. Bagher Rahimpour Cami made a content based movie recommendation system [19]. Content based system when merged with neural fuzzy approach gives a less effectiveness but when merged with a DNN can give a high quality recommendation system [20]. This type of recommendation system is better suited for recommending news, articles etc. because, while it can alleviate cold start issues by recommending new items to users, it is inefficient in terms of overspecialization because it recommends similar items but does not provide the user with newly updated items.

Collaborative filtering is primarily concerned with the user's rating [21]. When this strategy was introduced, it quickly gained popularity and was successfully used roughly twenty years ago. This strategy is based on the assumption that comparable consumers share similar interests. There are two variants of this technique: memory-based and model-based. Recommendation system development for fashion retail e-commerce is an application of extending collaborative filtering method [22]. Memory-based storage is further subdivided into user- and item-based storage [23]. It is mostly based on the algorithm of the nearest neighbour. Occasionally, the rating matrix is large, since it must deal with a large amount of data, and the solution time for the user-item rating matrix is lengthy. Additionally, it has disadvantages in terms of cold start issues and the inability to promote unpopular things to other users.

Moreover, it does not give real-time recommendations. A model-based recommendation system concatenates user and item attributes into a single factor space. It is primarily based on machine learning, which aids in the resolution of scalability and data sparsity issues.

Hybrid recommendation systems combine the best features of content-based and collaborative recommendation systems. In this manner, the disadvantages of both types are offset, making them more successful when employed. The items are initially sent into a content-based approach following collaborative filtering, which they refined using the k-means algorithm and a previously constructed user interest model. [24]. The Pro-FriendLink algorithm, based on the hybrid system, was presented to forecast movies utilizing CBRs, the DBScan clustering algorithm, and the DNN algorithm [25]. Deep learning based recommendation system – 'Pubmender' has its application in biomedical choices [26].

## 3. PROPOSED WORK

In our proposed system, we take several parameters and put them in an item- pool after analysing them using different algorithms after which they are re-ranked for increased efficiency. We have divided the model in five stages each of which is explained below.

### 3.1 First stage

All of the data from web log files is gathered and fed into a deep neural network model, in this case, the SOFTMAX model. A SoftMax is a function that gives us probabilities for various classes in a multi-class classification model. A SoftMax model calculates the relative chance of correctly predicting each item by treating each problem as a multi-class problem, allowing us to compare multiple items directly in a single model. In the SOFTMAX model, all of the fetched web log files and previous search information are fed into an information layer- a. We denote the product layer as  $\Psi(a) \in \mathbb{R}^f$ . The model converts the output of the last layer into a probability distribution by generating a vector of scores. The product layer uses the following probability distribution:  $\hat{p} = h(\Psi(a)V_z^U)$ , where:  $h: \mathbb{R}^n \rightarrow \mathbb{R}^n$  is the softmax function, given by  $h(x)_y = \frac{e^{x_y}}{\sum_z e^{x_z}}$ ,  $v \in \mathbb{R}^n \times \mathbb{R}^f$  is the matrix of weights of the softmax layer.

We introduce a LOSS function, that compares the probability distribution with all of the objects that the user has interacted with, which is represented as a multi-hot distribution.

The probability of the recommended item z is given by  $\hat{p}_z = \frac{\exp\{\Psi(a)V_z\}}{C}$ , where C is the normalization constant and does not depend on z. Here,  $\Psi(a) \in \mathbb{R}^f$  is the output of hidden layer and  $V_z \in \mathbb{R}^f$  is the vector of item connecting to product z. After this we do the data training which consists of two parameters, query features - a, and the data that user has interacted with - p. We use negative sampling in our model to avoid folding difficulties in our data training. It creates an approximate gradient by combining all positive entries in the target table with a random sample of negative things. Figure 1 shows the flow of the model. Finally, our data is now transmitted to an intermediate item pool after it has been processed.

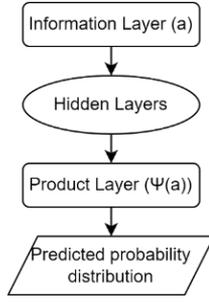


Figure 1. Softmax prediction

### 3.2 Second stage

A clustering method is now used to examine user behaviour that is comparable. All of the weblog files that have been acquired are first pre-processed to remove any unnecessary or noisy data. It then creates clusters, which are now known as information points taken as  $a, a(k)$ . Following that, we create  $k$  cluster centroids  $T$  and place  $ty \dots tk$  in various random locations. Each data point in the cluster is grouped according to its proximity to the centroid by using  $ty = \{z: f(az, \mu y) \leq f(az, \mu l), l \neq y, z=1, n\}$   $\mu y = 1/|ty| \sum z \in ty az, \forall l$  for  $\mu y =$  some value,  $y=1, k$ .

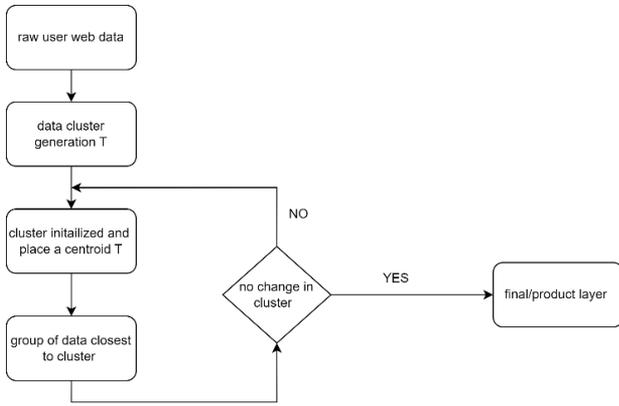


Figure 2. K-means clustering

Following the placement of centroid  $T$ , the nearest group to the cluster is examined for any changes to the cluster; if not, the initialization and placement of centroid  $T$  are repeated. If so, the product layer receives the finished result. This procedure is repeated until the convergence conditions are satisfied. Figure 2 shows the clustering iteration. The data is now transmitted to the item pool, where it will be re-ranked.

### 3.3 Third stage

Continuing with our model, we undertake cross-domain analysis, as well as user sentiment analysis, using a transfer learning method based on a multi-layer convolutional neural network i.e., CNN\_FT model. All of the data from the source (here, social media files) and target domains is sent into the CBOW (bag-of-words) neural network to train the words according to context. CBOW uses vector representation to train words from massive volumes of unstructured data to extract syntactic and semantic information. For doing so, it uses the given Eq. (1) for which value of  $\hat{z}$  is given by Eq. (2). The loss function for this algorithm is given as Eq. (3) for which the iterative formula is in Eq. (4) where  $\beta$  is the learning

speed. Moving on, the source domain data is first provided to the CNN-FT model in an information layer which is denoted by  $xy \in R_y^{m \times 1}$ . The original information layer is converted to  $xy \in R_y^{m \times k}$  using Word2vec after which the sentence is expressed as given in Eq. (5). The features of sentences are extracted in the following layer, the convolutional layer. To do so, we introduce  $h \times k$  in  $w_y \in R_y^{h \times k}$  in the information layer where the line is  $m - h + 1$ , in order to get the feature map, sentence for which is expressed  $c_y = (c_y^1, c_y^2, \dots, c_y^{m-h+1})$  for which  $c_y^i = f(w_y \cdot zy_{i:i+h-1} + b_y)$ . The data is subsequently delivered to the model's third layer, the pooling layer, which removes the most significant elements to improve accuracy. The main feature is defined by Eq. (6) which essentially contains the maximum value of the feature values. The last layer i.e., a fully connected layer uses the formula in Eqns. (7) and (8) to get the probability of each class. Furthermore, a small portion of the labelled data fields is used for the target domain and performs the procedure above until the pooling stage, after which the sentiment analysis is performed. The weights in the last layer of the fully connected layer are processed using the stochastic gradient descent method, which also allows adjusting the weight value. Figure 3 represents CNN\_FT flow chart. Our intermediary item pool receives the product.

$$Q = U \hat{z} \quad (1)$$

$$\hat{z} = \frac{1}{a \sum_{i=1}^a z_i} \quad (2)$$

$$L = -\log p(w_b | w_{b-a}, \dots, w_{b-1}, w_{b+1}, \dots, w_{b+a}) \quad (3)$$

$$W_b = w_b - \beta (\hat{z} - z_i), i \in (l, a) \quad (4)$$

$$zy_{1:m} = zy_1 \oplus zy_2 \oplus \dots \oplus zy_m \quad (5)$$

$$\max(c_y) = (\max(c_y^1), \max(c_y^2), \dots, \max(c_y^{m-h+1})) \quad (6)$$

$$\hat{s} = w \cdot \max(c_y) + d \quad (7)$$

$$P(\hat{s}_y^i) = \frac{\exp(\hat{s}_y^i)}{\sum_{i=0}^{label} \exp(\hat{s}_y^i)} \quad (8)$$

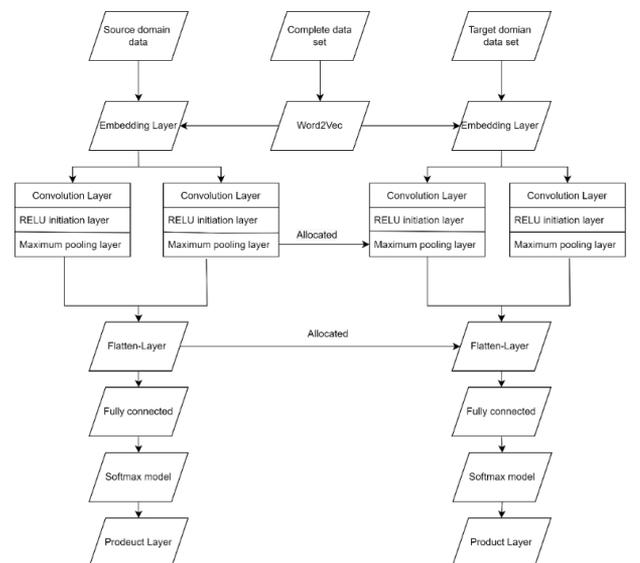


Figure 3. CNN\_FT for cross domain analysis

### 3.4 Fourth stage

We examine user feedback using EINMF model which is essentially a neural matrix factorization method based on explicit and implicit data of the user. The number of user and item are denoted as  $m$  and  $n$  respectively, for which we have datasets as  $P = \{p_1, p_2, \dots, p_m\}$  for users and  $Q = \{q_1, q_2, \dots, q_n\}$  for items. We construct the user-item explicit matrix  $T = [t_{pi}]_{m \times n}$  and implicit matrix  $IT = [it_{pi}]_{m \times n}$ . shown in Eqns. (9) and (10) which we use as the information in the neural matrix factorization model. After this inlaying is initialized to get the explicit-implicit feedback latent feature vectors  $x_p^{(E)}$  and  $x_p^{(I)}$  of users and the explicit-implicit feedback latent feature vectors  $y_i^{(E)}$  and  $y_i^{(I)}$  of items. We add the two vectors as shown in Eqns. (11) and (12) which serves as the information of the training layer to obtain shallow linear preference features of the user using the formula given in Eq. (13).

$$IT_{pi} = \begin{cases} 1, & \text{if interaction (user } p, \text{ item } q) \text{ is} \\ & \text{observed} \\ 0, & \text{un - interaction (user } p, \text{ item } q) \end{cases} \quad (9)$$

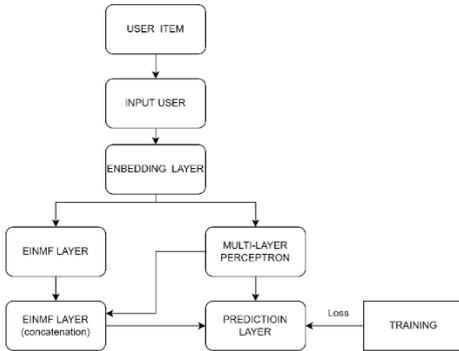
$$T_{pi} = \begin{cases} t_{pi}, & \text{if rating (user } p, \text{ item } q) \text{ is observed,} \\ 0, & \text{rating is unknown (user } p, \text{ item } q), \end{cases} \quad (10)$$

$$X = [x_i^{(E)} \oplus x_i^{(I)}] \text{ for user} \quad (11)$$

$$Y = [y_i^{(E)} \oplus y_i^{(I)}] \text{ for item} \quad (12)$$

$$\phi_{EI}^{dot}(X_p, Y_i) = X_p \theta Y_i \quad (13)$$

Proceeding with the model, we obtain the deep non-linear preference features by using the hidden multilayer perceptron (MLP) as shown in Eq. (14). (In this model ReLU is used as the activation function). We get the user liked item by using the function given in Eq. (15) as shown in Figure 4. The loss function in our model is the hybrid of explicit and implicit feedback given in (16). The output is then sent to our intermediate item pool for re-ranking.



**Figure 4.** EINMF algorithm for explicit and implicit data

$$z_{EI} = \begin{bmatrix} X_p \\ Y_i \end{bmatrix}$$

$$\phi_{EI}^1(z_{EI}) = c_{EI}^1 (W_{EI}^1 z_{EI} + b_{EI}^1)$$

$$\phi_{EI}^2(z_{EI}) = c_{EI}^2 (W_{EI}^2 z_{EI} + b_{EI}^2) \quad (14)$$

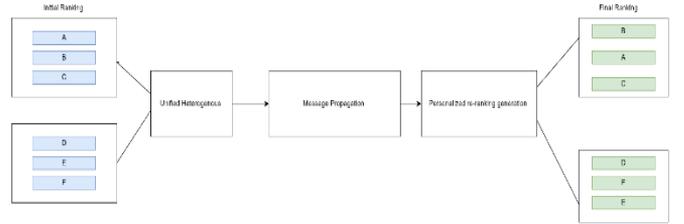
$$\phi_{EI}^R(z_{EI}^{R-1}) = c_{EI}^R (W_{EI}^R z_{EI}^{R-1} + b_{EI}^R)$$

$$\hat{k}_{pi}^{(EL)} = c_{out} (h^T \begin{bmatrix} \phi_{EI}^{dot} \\ \phi_{EI}^{MLP} \end{bmatrix}) \quad (15)$$

$$L = \eta L_I + (1 - \eta) L_E \quad (16)$$

### 3.5 Fifth stage

For re-ranking, all intermediate suggested items are passed to an item pool. We take a set of users  $P = \{p_1, p_2, \dots, p_p\}$ , items as  $\{Q = q_1, q_2, \dots, q_q\}$ , initial ranked list  $T(p)$  and a final generated list  $S(p)$  to be recommended to the user. Figure 5 shows the flow of our algorithm.



**Figure 5.** Re- ranking algorithm

We begin by constructing a unified heterogeneous graph, which is based on item relationships and user-item scoring. Item relationship graph  $G_I$ , which has item feature vector  $x_{q_i} \in \mathbb{R}^d$  for each item and edge feature vector  $e_{q_i, q_j} \in \mathbb{R}^c$  for connecting items  $q_i$  to  $q_j$ , show mutual influences among different items. While user-item scoring graph  $G_s$ , which have two types of nodes i.e., user and items to derive a relationship between user and item as each node consists of feature vector given as  $R_p \in \mathbb{R}^d$  and  $R_q \in \mathbb{R}^d$ .

Following that, we create several message propagation phases, divided into global item relationship propagation and customized intent propagation. The item and its feature vector are represented by initializing  $h^{(0)}_{q_i}$  by  $r_{q_i}$  in the  $G_I$  graph after which they are updated by two steps: message aggregation and message updating. A message aggregation step captures relational information from linked neighbours. The message is defined using a linear transformation from item  $q_i$  to  $q_j$  given by Eq. (17). The output is taken as a matrix of dimension  $d \times d$  given by Eq. (18), after which we take a mean aggregator to average the messages from neighbours by using Eq. (19). Followed by a message update step that fuses the information from the neighbours with the item's information using Gated Re-current Unit (GRU), Eq. (20). As a result, the item representation encodes relational and transitive connections across items across distinct ranked lists. Now, in order to improve the efficiency of our scoring, we're using a personalized intent propagation network that includes USERS as the centre node and their neighbours as items edges represent scores given by initial rankers. For each user  $p$ , the used features are loaded into a  $d \times d$  transformation matrix, which is then propagated up the user-item scoring graph.  $W_p = f_i(r_p)$  where  $f_i$  is non-linear mapping. In order to design a transformation matrix  $W_n \in \mathbb{R}^{d \times d}$  we introduce mapping by  $f_n(e_{qp}) = \text{MLP}(e_{qp})$  after this we combine nonlinear mapping from neighbours of user  $p$  to aggregate the messages Eq. (21). Ending with updating the user representation using  $h'_p = \text{GRU}(h_p, m_p)$ . Next, we concatenate user representation and item representation  $h_p$  and  $h_q$  respectively,  $q \in Q$ . This data is passed to a feed-forward neural network, which generates a re-ranking score for an item  $q$ .

$\hat{k}_{pq} = \sigma(MLP(h_p || h_q))$  where  $\sigma(r) = \frac{1}{1+e^{-r}}$  is the sigmoid function. We use BPR loss function given in Eq. (22) it ranks positive samples higher than negative samples.

This is our final score which will be then recommended to our end user as shown in Figure 6.

This is the proposed Stratified Advance Personalized Recommendation System based on Deep Learning (SAP Model).

$$m_{q_i \leftarrow q_j} = w_m h_{q_j} = f_m(e_{q_j q_i}, h_{q_j}) \cdot h_{q_j}, \quad (17)$$

$$f_m(e_{q_j q_i}, h_{q_j}) = MLP(e_{q_j q_i} || h_{q_j}) \quad (18)$$

$$m_{q_i} = \frac{1}{|N_{q_i}|} \sum_{q_j \in N_{q_i}} m_{q_i \leftarrow q_j} \quad (19)$$

$$h'_{q_i} = GRU(h_{q_i}, m_{q_i}) \quad (20)$$

$$m_p = \frac{1}{N_p} \sum_{p \in N_p} W_p W_n h_q = \frac{1}{N_p} \sum_{q \in N_p} f_l(x_p) f_m(e_{qp}) h_q \quad (21)$$

$$\mathcal{L} = \sum_{(p, q_i, p_j) \in a} \log \sigma(\hat{k}_{pq_i} - \hat{k}_{pq_j}) \quad (22)$$

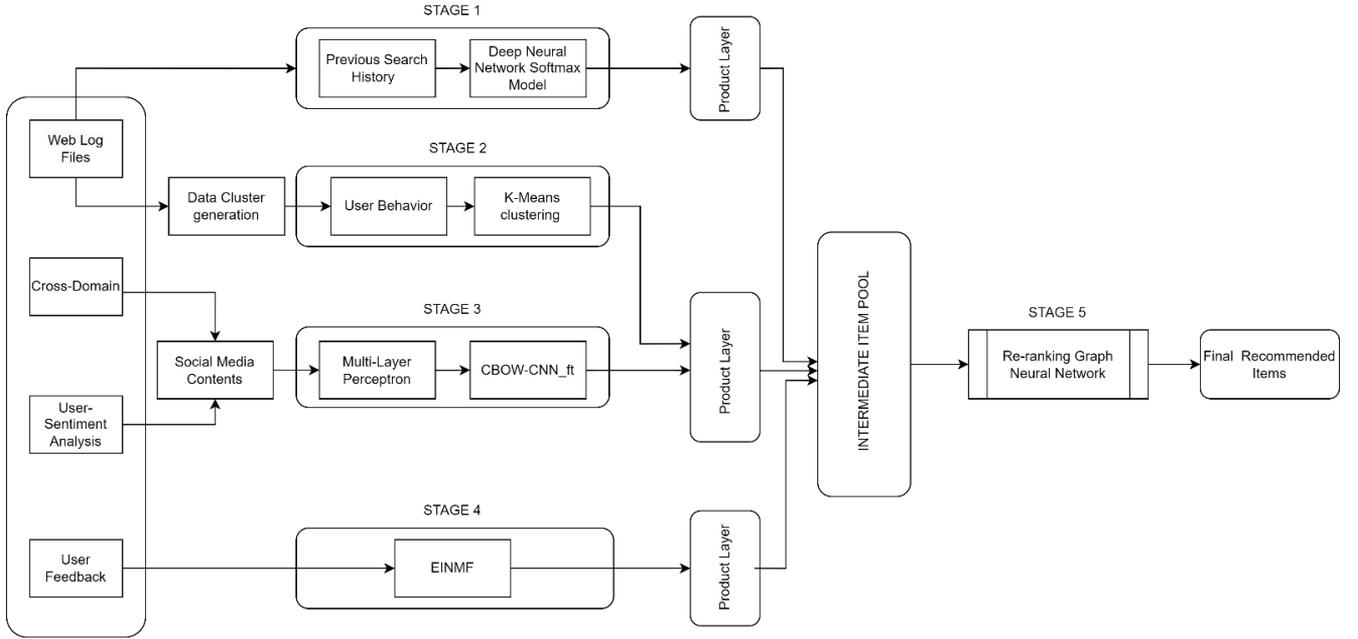


Figure 6. Proposed model diagram

This is the proposed Stratified Advance Personalized Recommendation System based on Deep Learning (SAP Model) (Figure 6). The proposed model is a deep learning-based recommendation system that is efficient for solving existing problems like cold start, and overspecialization. We also overcome data sparsity problems and try to increase the diversity of items to be recommended to the user. We employ a softmax layer which is used to predict the item the user is most likely to interact with. Continuing with the model. K-means is used to cluster users based on their item interactions or demographic information, allowing for more personalized recommendations to be made for each user segment. Then we use the CBOW-CNN\_FT method which is a word embedding method that can be used to capture the semantic relationships between items, allowing for more accurate item-item similarity measurements. Further, we take EINMF which helps solve several existing problems in recommendation systems by incorporating both explicit and implicit feedback, incorporating additional information, and leveraging the power of deep learning to generate more accurate and personalized recommendations. All the output layers are incorporated in a product layer which is then sent to an intermediate item pool. After collecting all the data it is then sent for re-ranking where we use a graph neural network for which the final ranking is assigned to each item. This

improves the suggested model's accuracy, efficiency, and quality of the proposed model.

## 4. EXPERIMENT AND RESULTS

### 4.1 Dataset

We take the amazon dataset as input in our model as shown in Table 1 [27]. It represents web log files and feedback of users along with social media contents.

Table 1. Amazon dataset

Category	User	Items	Reviews	Edges
Weblog files (Electonics)	4.25M	498K	11.4M	7.87M
User Feedback (Women's Clothing)	1.82M	838K	14.5M	17.5M
Social media content (Music)	1.13M	557K	6.40M	7.98M
All	7.20M	1.89M	31.90M	33.35M

### 4.2 Baseline

DEEPCONN model: This model uses user and item neural networks and takes their reviews as input, following layer is

CNN based which calculates various feature levels. They also introduce a top layer so that hidden latent features of user and item communicate. Finally, the output layer represents the recommended items [28].

IA CN model: In this model, user and item documents are taken as input and sent to an embedding layer which is a fully connected layer. Following that, it is sent to a CNN layer. They have an additional Attention Layer for effective output. Factorization is done next, after which the final output is recommended [29].

DMHR model: This study suggests a hybrid recommendation methodology based on multi-source perspective fusion and in-depth emotion analysis. The post-based collaborative filtering recommendation view and the content-based recommendation view are combined to create a hybrid recommendation system based on the analysis of user behaviour preferences, which focuses on the emotional mining and deep semantic analysis of text information; and the natural language description information of the post content is mined [30].

GHRS model: The basic concept behind the suggested approach is to identify connections between users based on their shared characteristics as nodes in a similarity network and then combine those connections with user-side data to address the cold-start problem [31].

### 4.3 Testing

We compare given models with our model for accuracy, precision and F- measure as shown in Figure 7.

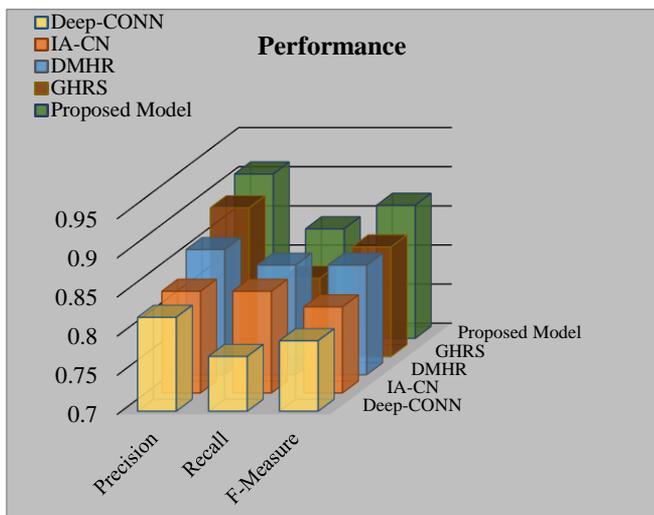


Figure 7. Precision, recall, F-measure comparison

#### 4.3.1 Precision

It is the ratio of True Positives to that of the sum of True Positives and False Positives. We calculate the percentage among the following models, DeepCoNN, IA CN, DMHR, GHRS and the proposed model. As shown in Table 2, the proposed model gives a precision of 91% compared to that of other models. Clearly, the proposed model shows higher precision.

#### 4.3.2 F-measure

It is the harmonic mean of two fractions given as:

$$F\text{-MEASURE} = \frac{PRECISION \times RECALL \times 2}{(PRECISION + RECALL)}$$

Table 2 shows the results, proposed model outstands at 87% compared to that of other models.

#### 4.3.3 Recall

It helps in knowing what percentage of real positives were successfully identified as shown in Table 2.

$$RECALL = \frac{TRUE\ POSITIVE}{TRUE\ POSITIVE + FALSE\ POSITIVE}$$

Table 2. Precision, Recall and F-Measure comparison in tabular form

Model	Precision	Recall	F-Measure
Proposed Model	0.91	0.84	0.87
GHRS	0.89	0.80	0.84
DMHR	0.86	0.84	0.84
IA-CN	0.83	0.83	0.81
DeepCONN	0.82	0.77	0.79

#### 4.3.4 MAE value

Mean absolute error (MAE) help's us to measure the average of the absolute error value. Absolute is a mathematical function that make a number positive. Therefore, the difference can be positive or negative and will be necessary positive when calculating MAE.

As shown in Figure 8, the proposed model as lowest MAE value.

$$MAE = \frac{(ACTUAL\ VALUE - PREDICTED\ VALUE)}{NUMBER\ OF\ VALUE}$$

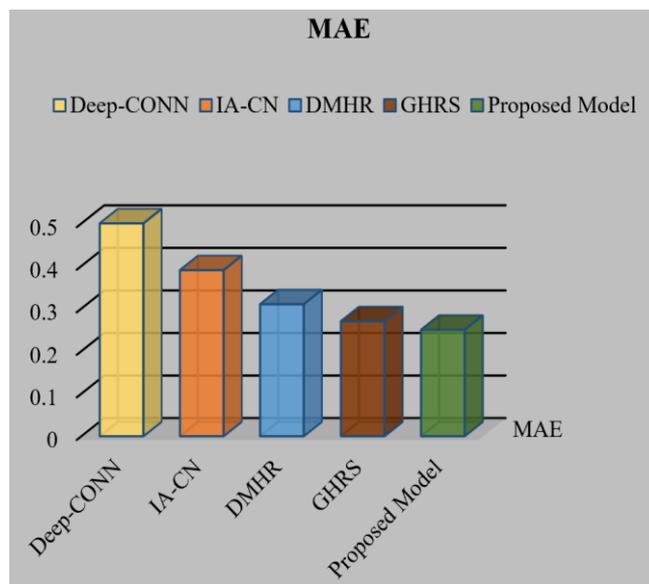
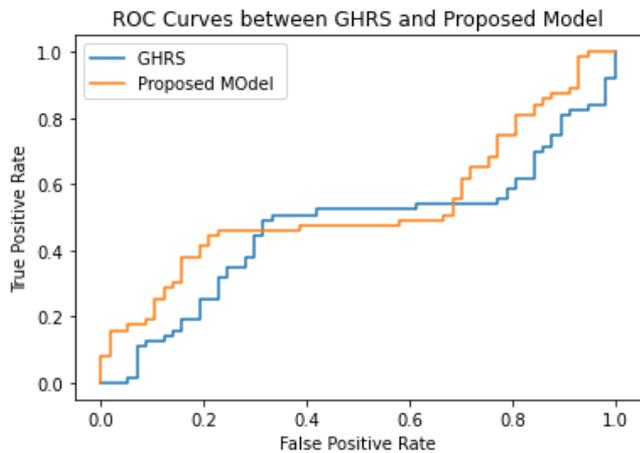


Figure 8. MAE value comparison

#### 4.3.5 AUC-ROC curve

A performance indicator for classification issues at different threshold levels is the AUC-ROC curve. AUC stands for the level or measurement of separability, and ROC is a probability curve.

From above comparison, we conclude that top two models are GHRS and the proposed model, for which we use AUC-ROC curve to determine the better model. From Figure 9 we conclude that the proposed model has more area under the curve and is therefore more efficient.



**Figure 9.** AUC-ROC curve

## 5. CONCLUSION AND FUTURE WORK

Recommendation system is a modern technology which helps in consumers locate goods they want to purchase as per their needs. Rapid advancements in recommendation system have made them an essential tool for online business. Deep neural networks are added to conventional recommendation models, they perform significantly better than linear methods like collaborative filtering and have higher efficiency. The suggested model effectively addresses the challenges of overspecialization and cold start. It can also process changes in user behavior and decision-making to overcome sparsity. Furthermore, it can address latency problems that occur when new products are posted yet older items are being recommended. We collect multiple criteria, such as browser log files, social media contents, and so on, and feed them through several relevant algorithms to build an intermediate item pool in which re-ranking occurs, and the final products are chosen to be suggested to our user.

We compared the suggested model to the DeepCONN, IA-CN models, DMHR and GHRS and the results are unambiguous: our model surpasses all for the provided dataset and has a minimum error with 91% accuracy. Additionally, we compared precision, recall, and F-measure, where the suggested model performed remarkably well. We also compare for MAE values and AUC-ROC and we conclude the proposed model is more efficient.

In addition to e-commerce, recommendation systems are extensively used in B2C and B2B business formats. The same might be searched for further in the suggested model. We would like to address the problems of high cost usually involved in deep learning based algorithms.

## REFERENCES

[1] Zhang, Q., Lu, J., Jin, Y.C. (2021). Artificial intelligence in recommender systems. *Complex & Intelligent Systems*, 7: 439-457. <https://doi.org/10.1007/s40747-020-00212-w>

[2] Neeley, V.S., Buluç, A., Gilbert, J.R., Olikier, L., Ouyang, W. (2016). Lira: A new likelihood-based similarity score for collaborative filtering. *ArXiv*, abs/1608.08646. <http://arxiv.org/abs/1608.08646>

[3] Li, L.H., Zhang, Z., Zhang, S.D. (2021). Hybrid

algorithm based on content and collaborative filtering in recommendation system optimization and simulation. *Scientific Programming Towards a Smart World 2021*. <https://doi.org/10.1155/2021/7427409>

[4] Gope, J., Jain, S.K. (2017). A survey on solving cold start problem in recommender systems. 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, pp. 133-138. <https://doi.org/10.1109/CCAA.2017.8229786>

[5] Khoali, M., Tali, A., Laaziz, Y. (2021). A survey of artificial intelligence-based E-commerce recommendation system. *Emerging Trends in ICT for Sustainable Development*, pp. 99-108. [https://doi.org/10.1007/978-3-030-53440-0\\_12](https://doi.org/10.1007/978-3-030-53440-0_12)

[6] Zhang, S., Yao, L.N., Sun, A.X., Tay, Y. (2020). Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys*, 52(1): 1-38. <https://doi.org/10.1145/3285029>

[7] Narayan, S. (2020). Multilayer perceptron with auto encoder enabled deep learning model for recommender systems. *Future Computing and Informatics Journal*, 5(2): 96-116. <http://doi.org/10.54623/fue.fcij.5.2.3>

[8] Ferreira, D., Silva, S., Abelha, A., Machado, J. (2020). Recommendation system using autoencoders. *Applied Sciences*, 10(16): 5510. <https://doi.org/10.3390/app10165510>

[9] Tercan, E., Atasever, U.H. (2021). Effectiveness of autoencoder for lake area extraction from high-resolution RGB imagery: An experimental study. *Environmental Science and Pollution Research*, 28: 31084-31096. <https://doi.org/10.1007/s11356-021-12893-y>

[10] Yang, D., Zhang, J., Wang, S.F., Zhang, X.D. (2019). A time-aware CNN-based personalized recommender system. *Hindawi Complexity*, 2019: 9476981. <https://doi.org/10.1155/2019/9476981>

[11] Rosewilt, L.A., Renjit, J.A. (2020). A content recommendation system for effective e-learning using embedded feature selection and fuzzy DT based CNN. *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, 39(1): 795-808. <https://dl.acm.org/doi/10.3233/JIFS-191721>

[12] Wu, C.H., Wang, J.W., Liu, J.T., Liu, W.Y. (2016). Recurrent neural network based recommendation for time heterogeneous feedback. *Knowledge-Based Systems*, 109: 90-103. <https://doi.org/10.1016/j.knosys.2016.06.028>

[13] Padmaja, S., Sheshasaayee, A. (2016). Clustering of user behaviour based on web log data using improved K-means clustering algorithm. *International Journal of Engineering and Technology (IJET)*, 8: 305-310. <https://www.enggjournals.com/ijet/docs/IJET16-08-01-095.pdf>

[14] Meng, J.N., Long, Y.C., Yu, Y.H., Zhao, D.D., Liu, S. (2019). Cross-domain text sentiment analysis based on CNN\_FT method. *Information*, 10(5): 162. <https://doi.org/10.3390/info10050162>

[15] Liu, H.Z., Wang, W., Zhang, Y.H., Gu, R.Q., Hao, Y.Q. (2022). Neural matrix factorization recommendation for user preference prediction based on explicit and implicit feedback. *Computational Intelligence and Neuroscience*, 2022(7): 1-12. <https://doi.org/10.1155/2022/9593957>

- [16] Liu, W.W., Liu, Q., Tang, R.M., Chen, J.Y., He, X.Q., Heng, P.A. (2020). Personalized re-ranking with item relationships for E-commerce. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 925-934. <https://doi.org/10.1145/3340531.3412332>
- [17] Sivapalan, S., Sadeghian, A., Rahanam, H., Madni, A.M. (2014). Recommender systems in E-commerce. 2014 World Automation Congress. <http://dx.doi.org/10.13140/2.1.3235.5847>
- [18] Tai, Y.F., Sun, Z.Y., Yao, Z.X. (2021). Content-based recommendation using machine learning. 2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1-4. <https://doi.org/10.1109/MLSP52302.2021.9596525>
- [19] Cami, B.R., Hassanpour, H., Mashayekhi, H. (2017). A content-based movie recommender system based on temporal user preferences. 2017 3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS), Shahrood, Iran, pp. 121-125. <https://doi.org/10.1109/ICSPIS.2017.8311601>
- [20] Rutkowski, T., Romanowski, J., Woldan, P., Staszewski, P., Nielek, R., Rutkowski, L. (2018). A content-based recommendation system using neuro-fuzzy approach. IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Janeiro, Brazil, pp. 1-8. <http://dx.doi.org/10.1109/FUZZ-IEEE.2018.8491543>
- [21] Zhang, R.S., Liu, Q.D., Gui, C., Wei, J.X., Ma, H.Y. (2014). Collaborative filtering for recommender systems. Second International Conference on Advanced Cloud and Big Data, Huangshan, China, pp. 301-308. <https://doi.org/10.1109/CBD.2014.47>
- [22] Hwangbo, H., Kim, Y.S., Cha, K.J. (2018). Recommendation system development for fashion retail e-commerce. Electronic Commerce Research and Application, 28: 94-101. <https://doi.org/10.1016/j.elerap.2018.01.012>
- [23] Sarwar, B., Karypis, G., Konstan, J., Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. Association for Computing Machinery, pp. 285-295. <https://doi.org/10.1145/371920.372071>
- [24] Li, L.H., Zhang, Z., Zhang, S.D. (2021). Hybrid algorithm based on content and collaborative filtering in recommendation system optimization and simulation. Scientific Programming, 2021(4): 1-11. <https://doi.org/10.1155/2021/7427409>
- [25] Farashah, M.V., Etebarian, A., Azmi, R., Dastjerdi, R.E. (2021). A hybrid recommender system based-on link prediction for movie baskets analysis. Journal of Big Data, 8: 1-24. <https://doi.org/10.1186/s40537-021-00422-0>
- [26] Feng, X.Y., Zhang, H., Ren, Y.J., Shang, P.H., Zhu, Y., Liang, Y.C., Guan, R.C., Xu, D. (2019). The deep learning-based recommender system “pubmender” for choosing a biomedical publication venue: Development and validation study. Journal of Medical Internet Research, 21(5): e12957. <https://doi.org/10.2196/12957>
- [27] McAuley, J., Pandey, R., Leskovec, J. (2015). Inferring networks of substitutable and complementary products. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. <https://doi.org/10.1145/2783258.2783381>
- [28] Zheng, L., Noroozi, V., Yu, P.S. (2017). Joint deep modelling of users and items using reviews for recommendation. Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, pp. 425-434. <http://dx.doi.org/10.1145/3018661.3018665>
- [29] Ming, F.P, Tan, L., Cheng, X.F. (2021). Hybrid recommendation scheme based on deep learning. Mathematical Problems in Engineering, 2021(01): 1-12. <https://doi.org/10.1155/2021/6120068>
- [30] Jiang, L., Liu, L., Yao, J.J., Shi, L.L. (2020). A hybrid recommendation model in social media based on deep emotion analysis and multi-source view fusion. Journal of Cloud Computing, 9(1): 57. <https://doi.org/10.1186/s13677-020-00199-2>
- [31] Darban, Z.Z., Valipour, M.H. (2021). GHRS: Graph-based hybrid recommendation system with application to movie recommendation. Expert Systems with Applications, 200: 116850. <https://doi.org/10.1016/j.eswa.2022.116850>